RESEARCH-ARTICLE

Online Learning from Mix-typed, Drifted, and Incomplete Streaming Features

SHENGDA ZHUO, Jinan University, Guangzhou, Guangdong, China

DI WU, Southwest University, Chongqing, China

YI HE, William & Mary, Williamsburg, VA, United States

SHUQIANG HUANG, Jinan University, Guangzhou, Guangdong, China

XINDONG WU, Hefei University of Technology, Hefei, Anhui, China

Open Access Support provided by:

Southwest University

Jinan University

William & Mary

Hefei University of Technology

Published: 08 September 2025
Online AM: 19 June 2025
Accepted: 30 May 2025
Revised: 19 March 2025
Received: 05 December 2024

Online Learning from Mix-typed, Drifted, and Incomplete Streaming Features

SHENGDA ZHUO, College of Cyber Security, Jinan University, Guangzhou, China
DI WU, College of Computer and Information Science, Southwest University, Chongqing, China
YI HE, William & Mary, Williamsburg, Virginia, USA
SHUQIANG HUANG, College of Cyber Security, Jinan University, Guangzhou, China
XINDONG WU, Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

Online learning, where feature spaces can change over time, offers a flexible learning paradigm that has attracted considerable attention. However, it still faces three significant challenges. First, the heterogeneity of real-world data streams with mixed feature types presents challenges for traditional parametric modeling. Second, data stream distributions can shift over time, causing an abrupt and substantial decline in model performance. Additionally, the time and cost constraints make it infeasible to label every data instance in a supervised setting. To overcome these challenges, we propose a new algorithm *Online Learning from Mix-typed, Drifted, and Incomplete Streaming Features* (OL-MDISF), which aims to relax restrictions on both feature types, data distribution, and supervision information. Our approach involves utilizing copula models to create a comprehensive latent space, employing an adaptive sliding window for detecting drift points to ensure model stability, and establishing label proximity information based on geometric structural relationships. To demonstrate the model's efficiency and effectiveness, we provide theoretical analysis and comprehensive experimental results.

CCS Concepts: • **Computing methodologies** → **Online learning settings**;

Additional Key Words and Phrases: Online Learning, Mix-Typed, Streaming Feature

Associate Editor: Xingquan Zhu

This work was supported in part by the New Chongqing Youth Innovation Talent Project under Grant CSTB2024NSCQ-QCXMX0035, the National Natural Science Foundation of China under Grant (62176070 and 62272198), the National Key Research and Development Program of China under Grant 2024YFF0908200, the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China) under Grant BigKEOpen2025-03, the Guangdong Key Laboratory of Data Security and Privacy Preserving (2023B1212060036), the Guangdong-Hong Kong Joint Laboratory for Data Security and Privacy Preserving (2023B1212120007), the Guangdong Basic and Applied Basic Research Foundation (2024A1515010121), and the Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation (Climbing Program Special Funds) under Grant pdjh2025ak028. Y. He was not supported by any of these fundings.
Authors' Contact Information: Shengda Zhuo, College of Cyber Security, Jinan University, Guangzhou, China; e-mail: zhuosd96@gmail.com; Di Wu (corresponding author), College of Computer and Information Science, Southwest University, Chongqing, China; e-mail: wudi.cigit@gmail.com; Yi He, William & Mary, Williamsburg, Virginia, USA; e-mail: yihe@wm.edu; Shuqiang Huang, College of Cyber Security, Jinan University, Guangzhou, China; e-mail: hsq@jnu.edu.cn; Xindong Wu, Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China; e-mail: xwu@hfut.edu.cn.

ACM 1556-472X/2025/9-ART150
https://doi.org/10.1145/3744712

ACM Reference format:
Shengda Zhuo, Di Wu, Yi He, Shuqiang Huang, and Xindong Wu. 2025. Online Learning from Mix-typed, Drifted, and Incomplete Streaming Features. *ACM Trans. Knowl. Discov. Data.* 19, 8, Article 150 (September 2025), 28 pages.
https://doi.org/10.1145/3744712

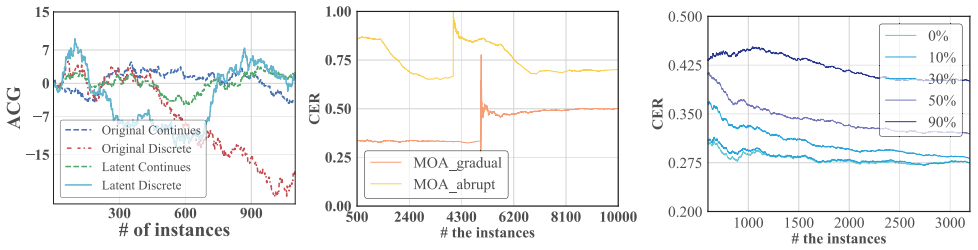
1 Introduction

Online learning from doubly streaming inputs has recently emerged as a cutting-edge paradigm in data stream analytics [2, 16, 22, 30, 39, 57, 60]. Unlike traditional online learning that considers data streams within a fixed feature space only [1, 43, 52], online learning with doubly streaming paradigm aims to adapt to the dynamic changes in both streaming data and streaming features. The new learning paradigm offers a more flexible learning environment, where new features can take arbitrary forms, and old features may become unobservable or disappear over time, which makes model training both more challenging and demanding.

Driven by this flexibility, various applications in different fields are gradually moving away from the fixed online learning paradigm and adopting the doubly stream paradigm to model their data [36, 38, 42, 48, 58]. The doubly streaming property emerges from the crowd-sensed data streams. When new users join the sensing effort with upgraded or entirely new devices (e.g., cell phones), they introduce new features. Conversely, when users depart or their devices fail to transmit data due to network issues, this can result in unobservable features. Prior research on these data streams typically shares a common goal: finding connections between features to create incremental models. This is often accomplished through two primary methods—(1) initializing learning coefficients for new features with informed guesses and addressing data sparsity issues and (2) pre-learning reconstruction information for unobserved features while leveraging previously learned parameters to enhance model performance.

Despite the success of the incremental models, most existing research is constrained by three specific assumptions. *First*, it is common in model training to assume that the streaming data features have the same data type [65]. However, this assumption contradicts real-world applications and overly idealizes the feature distribution in the input data. For example, financial institutions process massive amounts of transaction data from diverse sources, such as bank transactions, credit card payments, and mobile payments, which come in heterogeneous formats and require adaptive models capable of handling mixed data types effectively. *Second*, the phenomenon of concept drift, characterized by alterations in the probabilistic distribution of data, has been observed to have detrimental impacts on the performance of machine learning models [63]. This occurs as the models, trained on previous data, fail to accurately predict new instances of data that are drawn from a distinct distribution. This phenomenon is most common in manufacturing, where sensor data distributions can shift over time due to changes in machine usage, environmental conditions, or wear and tear, requiring robust models that adapt to concept drift. *Third*, the incremental learning mode in online learning is fully supervised, which means that the training process requires every data instance to have a class label. Unfortunately, having complete labels for a large amount of data streams is overly idealistic and comes with significant human and time costs. Moreover, in the cybersecurity phenomenon, intrusion detection systems analyze network traffic for anomalies, but the vast volume and complexity of incoming data make full labeling impractical, necessitating effective models that can handle partially labeled or unlabeled data to continuously improve detection accuracy.

Motivated by this situation, we introduce an approach to enhance the flexibility and applicability of doubly streaming data analytics, known as OL-MDISF. This new challenge encompasses three



(a) Challenge I (b) Challenge II (c) Challenge III

Fig. 1. Visualization of the three challenges. (a) Trends in the ACGs of the model. (b) Impact of different concept drift data on the CER of the model. (c) Influence of varying label scarcity on the CER of the model.

key problems (Figure 1), which will be collectively addressed in this article: (1) Scarce availability of labels for incoming data instances; (2) Concept drift, which diminishes the performance and accuracy of online learners as they struggle to adapt to shifting data distributions, and (3) An unbounded and evolving feature space that includes Boolean, ordinal, and continuous features simultaneously.

To tackle the aforementioned problem, our approach involves deriving a latent space from mix-typed streaming features while preserving the underlying geometric structure that imparts significant discriminative power to the data instances. To achieve it, our approach encompasses three crucial components: (1) The **Gaussian Copula (GC)** model captures the generative marginals of data by employing a set of latent and continuous probability densities, establishing correlations between them, (2) An online adaptive window model has been proposed to detect differences in data distribution under different sub-windows to prevent drift and resultant model collapse, and (3) An online adaptive density-peak clustering model that detects geometric relationships behind data instances to establish channels for propagating labeled information. We summarize our main *contributions* as follows.

- —This is a novel study to explore the online learning problem with mix-typed streaming features, data drifting, and semi-supervised. Its three challenges, namely how the mixed data types, the concept drifted data, and label scarcity can negatively affect the online learning efficacy, are presented in Section 3.
- —The algorithm to resolve the new OL-MDISF problem with copula modeling, adaptive slide windowing, and adaptive density-peak clustering is proposed and elaborated in Section 4. Its detailed analyses and descriptions are provided in Section 4.5.
- —A theoretical study substantiates (1) the tightness of online estimated statistics via our copula model ($cf.$, Lemmas 5.1 and 5.2) and that (2) our online learner enjoys a sub-linear regret bound ($cf.$, Theorem 1). Experiments benchmarked on 20 datasets evidence the viability, effectiveness, and superiority of our proposed algorithm with findings documented in Section 6.

The remaining sections of the article are structured as follows. In Section 2, we provide a concise overview of the relevant literature. We introduce the OL-MDISF learning problem, including the three challenges of this work and our ideas about it in Section 3. Then we introduce the OL-MDISF model, algorithmic logic (pseudocode), and complexity in Section 4. The experimental results and analysis in Section 6. Finally, the conclusion is drawn in Section 7.

2 Related Work

Online learning is a learning paradigm that operates by processing data instances as they arrive one by one. In theory, online learning has the capability to adapt to concept drift due to its real-time model update strategy. However, in the case of doubly streaming data and label scarcity, current online learning methods lack specific mechanisms and strategies for handling mix-typed data streams. This study explores a new online learning problem, namely OL-MDISF, which we relate to three research directions: *online learning from doubly streaming data*, *online concept drift detection*, and *online semi-supervised learning*. By introducing adaptive strategies for doubly streaming data, we unify them into these three key directions. This section provides an overview of the relevant prior research for each objective and discusses its respective limitations.

2.1 Online Learning from Doubly Streaming Data

Traditional online learning models for fixed data streams have certain limitations in meeting real-world demands. The recent emergence of the doubly streaming data online learning paradigm, by allowing the use of non-fixed feature spaces, has advanced the field of online learning, making it better suited for complex real-world applications. And [2, 16, 17, 21–24, 59] have also explored situations where features that emerged in previous rounds can become unobservable. LFES [22] explores the intricate relationship between vanishing and enduring features in evolvable data streams. However, the alignment of these evolvable data streams with practical contexts remains notably constrained. Operating within the dynamic landscape of variable feature spaces, OLVF [2] simultaneously navigates the realms of feature space and instance classification. Nevertheless, it regrettably overlooks the potential existence of heterogeneous data types within these adaptable feature spaces. The synergy between proactive query strategies and passive-aggressive update mechanisms, as exemplified by the PAATS [32] paradigm, finds its niche within the realm of online learning. Yet, its efficacy in addressing the challenges posed by imperceptible and antiquated data features remains to be fully realized. These studies establish a highly flexible and, thus, practical learning environment, given that it is often impractical to predefine a set of informative features and assume their consistent availability over extended time periods. They share a common technique, which involves establishing associations between new and old features. Even when old features cannot be directly observed, they can assist learners in training on new features by reconstructing their information, thereby improving prediction accuracy. The new features often have limited information due to the insufficient availability of data instances for model training.

Regrettably, previous studies predominantly operate under the assumption of a fully supervised learning context. Without labels, online learners cannot be updated efficiently, causing the slow acquisition of feature correlations, and consequently leading to the development of inadequately trained classifiers and inaccurately reconstructed features. This situation can result in significant prediction errors. In our OL-MDISF problem, our objective is to construct precise online learners capable of operating with limited labels, thus surpassing previous methodologies and achieving a higher level of practicality.

2.2 Online Concept Drift Detecting

Online concept drift detection improves the performance of online learners and is centered on effectively avoiding the effects of drift on the learner [29, 33, 37]. This damage is visible, such as abrupt changes, which lead to sudden changes in model loss, or invisible, such as gradual changes, where the model gradually loses sensitivity to old data. Drift points in the data stream are detected, enabling the model to take effective measures to avoid the harm caused by drift,

such as using two different update strategies [27, 48] that integrate the model under old and new data. OS-ELMs [61] address the issue of conceptual drift through the development and continuous updating of an online sequential extreme learning machine. This process involves quantifying the degree of modification that the updated model undergoes when exposed to newly collected data. However, the framework does not account for the intricacies posed by more complex scenarios involving dual-stream data. WIDSVM [13] leverages an incremental-decremental SVM algorithm in conjunction with vector migration conditions. Nonetheless, the SVM's innate capacity to generalize information prevents it from adequately adapting to the demands of more intricate data stream types. CPP [44] takes a dynamic approach by computing class *a posteriori* probabilities in real-time, aiming to capture evolving concepts. Nevertheless, its efficacy remains constrained when applied to real-world scenarios that demand true real-time performance.

The main challenge at present is that the concept drift detection does not consider the scenario of doubly stream input, and there is some data incompleteness. This requires the learning of data change distributions with guaranteed time dimension growth, while OL-MDISF considers the difference in distributions of old and new data under dual-stream data using a model with a double sub-window.

2.3 Online Semi-supervised Learning

Online semi-supervised learning enables models to progressively learn from the evolving geometric structure of the data stream, improving performance while reducing the need for labeled data. This structure can take two forms: explicit, such as topological spaces [25, 47, 51, 56], and implicit, such as Riemannian manifolds [12, 14, 28] or clustering structures [11, 15, 55]. Online learners can leverage these geometric structures to accelerate convergence, for example, by encouraging nearby instances to share the same labels. BLS [25] combines online kernel learning to achieve streamlined online updates. Nevertheless, it does not account for potential drift issues within the data. TLP [47] undertakes the continuous maintenance of a dimensional matrix for a data stream. It engages in real-time updates and learning for each incoming data point, although this maintenance process results in the partial loss of data information. OVSIS [18] establishes a universal feature space for the purpose of learning and quantifying the similarity between features. However, it is worth noting that the applicability of this universal feature space is limited to a narrow range of original data types.

Nonetheless, there have been limited semi-supervised online learners designed specifically for doubly streaming inputs. The primary challenge stems from the absence of a metric for equitably assessing the distance between pairs of data instances described by distinct feature spaces. This gap is what our OL-MDISF strives to investigate and bridge. To address this challenge, we propose utilizing the copula model to align feature spaces, thereby establishing relationships among diverse data types, including Boolean, ordinal, and continuous. This approach renders the measurement of the distance between data instances arriving over time feasible and equips our online learner to operate effectively with limited labels.

3 The OL-MDISF Learning Problem

Consider an input sequence $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$, where $\mathbf{x}_t \in \mathbb{R}^{d_t}$ represents a data vector with a dimension of d_t. In a doubly streaming scenario, we have $d_t \neq d_i$ for any two different rounds $t \neq i$ in general. With mix-typed streaming features, we express $\mathbf{x}_t$ as $\mathbf{x}_t := (\mathbf{x}_C, \mathbf{x}_D)$, where the subscripts C and D indicate the continuous and discrete (i.e., Boolean or ordinal) variables, respectively.

At each round t, the online learner f_t observes an instance $\mathbf{x}_t$ and provides its prediction $f_t(\mathbf{x}_t)$. With only a low probability, the true label $y_t \in \{-1, +1\}$ is disclosed, resulting in a loss $\ell(y_t, f_t(\mathbf{x}_t))$ for the learner. Using this loss information, the learner adapts to become f_{t+1} and prepares for the subsequent round. Our goal is to minimize the empirical risk:

$$R(T) = \frac{1}{l} \sum_{t=1}^{T} \pi(t) \cdot \ell(y_t, f_t(\mathbf{x}_t)), \tag{1}$$

where l denotes the total number of *labeled* instances over T rounds. $\pi(t)$ is an indicator function with $\pi(t) = 1$ for the rounds that reveal label y_t and with $\pi(t) = 0$ otherwise.

3.1 Challenges and Our Ideas

The formulation of the OL-MDISF problem reveals three key challenges, which are outlined as follows:

(1) *CH I—Mix-typed Features*: While the first-order oracle, specifically the gradient [6], is a commonly used and powerful optimizer for Equation (1), it's worth noting that data of various types often have features that span different value scales. As a result, the gradients derived from these diverse features can be muddled. For instance, the updating steps recommended by discrete features may operate at a coarser level of granularity than those generated by continuous features, resulting in more drastic updates.

To visualize, Figure 1(a) presents a toy example adapted from the "real-stream" dataset in Section 6, demonstrating the impact of feature types on gradient derivatives. Specifically, we observe that the **Average Cumulative Gradient (ACG)** [41] associated with discrete features exhibits more significant fluctuations than that of continuous features. The greater the ACG variation, the slower the coefficient learning speed for that feature. It's important to note that new features continuously emerge, and initializing their coefficients randomly or at zero can cause shifts in the decision hyperplane. Discrete features with high-gradient variations cannot provide meaningful optimal updates, leading to the inability to correct initialization biases. This results in more prediction errors by online learners.

(2) *CH II—Data Drifting*: The risk (loss) is in gradual convergence, and sudden conceptual drift occurs, which can lead to a precipitous rise in risk (loss). More intuitively, the online learner (*cf.*, $f(\cdot)$ in Equation (1)) can promise to relearn the new data distribution, but it takes more rounds to update it.

Figure 1(b) visualizes this self-awareness, using the experimental "MOA Abrupt" dataset as an example, which presents the damage caused by conceptual drift. The abrupt change in the data distribution can induce a rapid and substantial decrease in the efficacy of the machine learning model, as it fails to adjust to the sudden alteration promptly. Conversely, gradual changes in the data distribution result in a more gradual decline in performance and accuracy, necessitating ongoing updates to sustain the model's relevancy in the face of evolving data distributions.

(3) *CH III—Label Scarcity*: In rounds without labeled data, there is no risk (loss), so no gradients are computed to update the learner, as described in Equation (1). Intuitively, depending on the scarcity of labels, online learners can promise a certain convergence speed, implying that more rounds are needed for convergence.

Figure 1(c) visually illustrates this intuition, where the learner employs online convex programming. The **Cumulative Error Rate (CER)**, measuring the learner's predictive performance, is shown in the curve. We can observe that with increasing label scarcity, the CER curve tends to flatten, indicating slower convergence. It's worth noting that in an online learning scenario, learners observe data instances only once. In comparison to viewing the entire global data and achieving

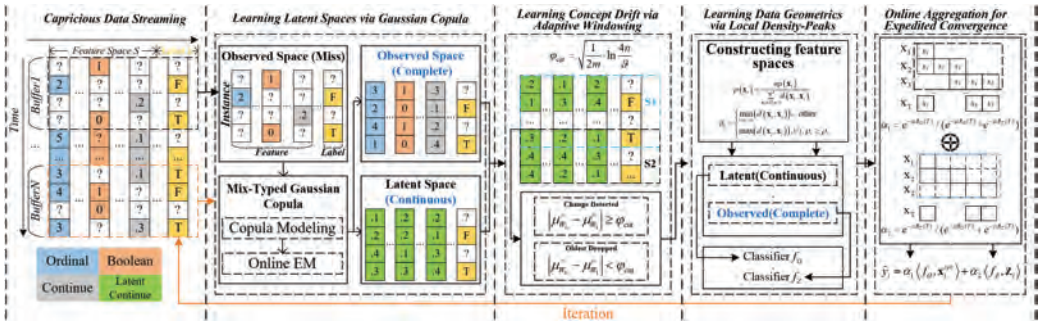


Fig. 2. Flowchart of our OL-MDISF model. The input data are mix-typed streaming features. (a) Learning Latent Spaces via GC is to construct the mixed data's Latent feature space and obtain the entire observed space. (b) Learning Concept Drift via Adaptive Windowing is to adjust the offset point of the sub-window detection data. (c) Learning Data Geometrics via Local Density-Peak is discovering the structure of data space based on a finding of local density peaks. (d) Online Aggregation for Expedited Convergence is an ensemble strategy of a classifier to improve the model's performance.

optimal prediction errors, a lower convergence rate for online learners implies higher prediction errors.

(4) *Our Ideas*: To overcome these three challenges, we briefly introduce three key ideas that motivated the design of our model (Figure 2). *First*, to control aggressive updates caused by mix-typed features, we aim to have a model that instantaneously normalizes oscillating gradients on discrete variables into the continuous domain. To achieve this, we propose the GC [19, 31] model, which can model complex multivariate distributions of mix-typed data streams, including the range of continuous normal variables and the dynamic distribution of discrete variables in a latent space. By training in this latent space, online learners can capture correlations between features and have two advantages: (1) the ability to initialize any new feature with informed guesses rather than purely random initialization (random initialization may introduce bias); and (2) any unobserved features can be reconstructed, enhancing prediction accuracy by leveraging their learned coefficients.

Second, to overcome the concept of drifted streams, we present the development of a novel dynamically adaptable window for the purpose of identifying instances of value drift. Our approach involves the implementation of two distinct sliding windows, which allow for the independent acquisition of both old and new data. Value drift is identified when the discrepancy between the mean values of the two sub-windows exceeds a predetermined threshold. Notably, this threshold is iteratively optimized via an adaptive adjustment in response to the distribution of the window values. Specifically, the distance between the cluster centroid and the farthest point within a given category is utilized to inform this adaptive thresholding process. Overall, the proposed methodology provides a robust and effective means for detecting value drift in a variety of applications.

Third, to address the label scarcity, we leverage a substantial amount of unlabeled data instances to infer the geometric structure of the input sequence. In this structure, instances with similar labels are clustered in adjacent regions, while instances with different labels may be scattered in distant regions. To uncover this geometric structure, we measure the distances between instance pairs in the latent space learned by the GC model and utilize their label relationships as a regularization term. We will formalize these two ideas into a regularization risk minimization mechanism and customize its objective function in the next section.

Table 1. Notations and Descriptions Used in OL-MDISF

Symbol	Description
x_t	Mix-typed input data instance at round t
y_t	Label of instance x_t (disclosed with low probability)
f_t	Online learner at round t
$R(T)$	Empirical risk over T rounds with labeled instances
$\pi(t)$	Indicator function (1 if y_t is revealed, 0 otherwise)
U_t	Union of all features observed up to round t
x_C	Continuous features of input x
x_D	Discrete (Boolean or ordinal) features of input x
z_t	Latent representation of x_t in GC
Σ	Covariance matrix in GC model
g	Monotonic transformation function in GC
x_O	Observed part of x_t
x_M	Missing (unobserved) part of x_t
z_O	Latent vector for observed features
z_M	Latent vector for missing features
$\tilde{z}_M$	Estimated latent vector for missing features
x_t^{rec}	Reconstructed input x_t from latent vector
$\mathcal{B}$	Buffer storing recent instances
φ_{cut}	Threshold for concept drift detection
ρ_t	Local density of instance x_t
δ_t	Distance to nearest higher-density instance
d_{cut}	Cutoff distance for density-based clustering
f_O	Classifier trained on original (observable) features
f_Z	Classifier trained on latent space features
y_O	Prediction made by f_O
y_Z	Prediction made by f_Z
α_1, α_2	Ensemble weights for f_O and f_Z
$R_O(T), R_Z(T)$	Cumulative risks for f_O and f_Z
μ	Learning rate for ensemble weight updates

4 The Proposed OL-MDISF Approach

Our approach can be conceptually framed into the objectives taking the following formulations:

$$\min_{f_1,\ldots,f_T} R(T), \quad \text{s.t. } \mathbf{x}_t \overset{\text{i.i.d.}}{\sim} \text{GC}(\mathbf{z}_t; \text{g}; \Sigma), \tag{2}$$

$$\max_{\text{g};\Sigma} \text{P}[\mathbf{x}_t \mid \mathbf{z}_t; \text{g}^{-1}, \Sigma], \quad \forall \mathbf{x}_t \in B. \tag{3}$$

The aim of Equation (2) is to minimize the risk of semi-supervised learning. It assumes that the input sequence $\{\mathbf{x}_t\}_{t=1}^T$ is independently sampled from an unknown distribution modeled by the GC. Equation (3) involves using an online **Expectation-Maximization (EM)** process to estimate the parameters of GC within the buffer B. Its goal is to find the latent representation $\mathbf{z}_t$ for each input $\mathbf{x}_t$ that contains mixed variable types, with this representation being composed of continuous normal variables. This section delves into the details of the model by thoroughly examining these objectives one by one. We define the notations (Table 1) used in this article.

4.1 Learning Latent Spaces via GC

The GC possesses the ability to model the joint distribution of mix-typed features and provides two critical properties. *First*, before entering round t ($t > 1$), we define $\mathcal{U}_t = \bigcup_1^t \mathbb{R}^t$, representing a common feature space that includes all features observed up to that point. Due to the dynamic changes in the feature space, each input $\mathbf{x}_t$ contains a subset of $\mathcal{U}_t$, where unobserved features may result in information loss, impacting learning efficiency. To address this challenge, GC maps the observed inputs to a latent space, which contains enough statistical information to estimate unobserved features. This reconstructed information provides support for learners, enabling them to make more accurate online predictions through aggregation methods.

Second, GC tames the garbled gradients by its definition:

Definition 4.1 (GC [34]). For a random vector $\mathbf{x} \in \mathbb{R}^d$ that adheres to the GC $(\mathbf{x}, \mathrm{g}, \Sigma)$, there exists a correlation matrix Σ and an element-wise monotone function $\mathrm{g} : \mathbb{R}^d \mapsto \mathbb{R}^d$ such that $\mathbf{x} = \mathrm{g}(\mathbf{z})$, where $\mathbf{z} \sim N_d(\mathbf{0}, \Sigma)$.

As observed, the latent representation of the input $\mathbf{x}_t := (\mathbf{x}_C, \mathbf{x}_D)$ consists of a collection of normally distributed continuous variables $\mathbf{z}_t$ with a mean of zero and a covariance matrix Σ. Training learners on the latent representations replaces the drastic updates proposed by discrete features with continuous gradients, facilitating a finer-grained search for minimization.

Combining the two aforementioned properties, we will introduce the definition of a monotonic function $g(\cdot)$ and the covariance matrix Σ. We employ a monotonically truncated operator [16, 62] for probability mass functions to estimate discrete variables in $\mathbf{x}_t$, where Σ remains invariant under strictly monotonic transformations of its elements. For a discrete feature x_i in $\mathbf{x}_D$ of size $|k|$, and a probability mass function $\{p_l\}_{l=1}^k$, the mapping is:

$$g_i := \mathrm{cutoff}(\mathbf{z}; \mathrm{S}) = 1 + \sum_{s_l \in \mathrm{S}} \mathbb{1}(z > s_l), \tag{4}$$

where $\mathbf{z} \in \mathbb{R}$ is continuous normal with **Cumulative Distribution Function (CDF)** $F_{\mathbf{z}}$ and $\mathrm{S} = \left\{ s_l = F_{\mathbf{z}}^{-1}(\sum_{t=1}^{l} p_t) : l \in |k-1| \right\}$. The latent vector is thus as $\mathbf{z}_t := \mathbf{f}^{-1}(\mathbf{x}_t) = (\mathbf{f}^{-1}(\mathbf{x}_C), \mathrm{cutoff}^{-1}(\mathbf{x}_D))$, so by the invertibility of monotone mappings. The latent representations for continuous or discrete features are either specified as real values or can be obtained from the Cartesian product of an interval, respectively.

Unobserved Feature Reconstruction. It's important to note that the dimension of $\mathbf{f}^{-1}(\mathbf{x}_t)$ is the same as that of $\mathbf{x}_t$ but may not match the dimension of $\mathcal{U}_t$. In our OL-MDISF problem, any feature can become unobservable, resulting in missing entries $\mathbf{x}_M \in \mathcal{U}_t \setminus \mathbb{R}^{d_t}$. To simplify notation, we represent the observed instance $\mathbf{x}_t$ as $\mathbf{x}_O$. In order to achieve a comprehensive latent representation, the objective is to reconstruct $\mathbf{z}_t = \phi(\mathbf{x}_t) \in \mathbb{R}^{|\mathcal{U}_t|}$ by establishing relationships between $\mathbf{x}_O$ and $\mathbf{x}_M$. Our solution involves mapping the marginal values of the observed $\mathbf{x}_O$ to $\mathbf{z}_M$ using conditional mean vectors. This feature reconstruction involves two approximation steps, namely: (1) Calculating the expectation of the observed $\mathbf{z}_O$ given the observation $\mathbf{x}_O$. (2) Determining the expectation of the missing $\mathbf{z}_M$ given $\mathbf{z}_O$, formulated as:

$$\begin{aligned} \tilde{\mathbf{z}}_M &= \mathbb{E}\Big[\mathbb{E}[\mathbf{z}_M \mid \mathbf{z}_O, \Sigma] \mid \mathbf{x}_O, \Sigma\Big] \\ &= \Sigma_{M,O} \cdot \Sigma_{O,O}^{-1} \cdot \mathbb{E}\left[\mathbf{z}_O \mid \mathbf{x}_O, \Sigma\right], \end{aligned} \tag{5}$$

where $\Sigma_{M,O}$ and $\Sigma_{O,O}$ represent the sub-matrices of the correlation matrix Σ with rows and columns corresponding to the feature indices of $(\mathbf{x}_M, \mathbf{x}_O)$ and $(\mathbf{x}_O, \mathbf{x}_O)$, respectively. Assuming $\tilde{\mathbf{z}}_M$ is an unbiased estimation of $\mathbf{z}_M$, we obtain a complete view of the latent representation as

$\mathbf{z}_t = (\mathbf{z}_O, \tilde{\mathbf{z}}_M)$. Hence, we can create a reconstructed version of the input $\mathbf{x}_t$ by sampling from the copula $\mathrm{GC}(\mathbf{z}_t, \mathbf{f}, \Sigma)$, denoted as $\mathbf{x}_t^{\mathrm{rec}} = (\hat{\mathbf{x}}_O, \hat{\mathbf{x}}_M) \in \mathcal{U}_t$.

Parameter Estimation. Through feature reconstruction, we can optimize the function $\mathbf{f}$ and correlation Σ in a stochastic and online manner by assessing the differences between the observed values $\mathbf{x}_t := \mathbf{x}_O$ and the reconstructed values $\hat{\mathbf{x}}_O$. To achieve this, we first define $g_i^{-1} = \Phi^{-1} \circ F_i$, where Φ represents the CDF of the standard normal distribution, while F_i corresponds to the CDF of the actual but unknown ith feature. We use the buffer B of input instances to empirically estimate F_i and obtain $\hat{F}_i$, allowing us to infer the distribution of continuous features. This process enables us to effectively grasp distribution information about the features, providing robust support for subsequent online learning. The estimator for the continuous feature is defined as follows:

$$\hat{g}_i^{-1}(x_i) = \Phi^{-1}\big(H \cdot \hat{F}_i(x_i)\big), \tag{6}$$

where the scaling factor $H = |B|/(|B|+1)$ is introduced to ensure the boundedness of the output values. For discrete features, we can consider the truncation values S^i as a special case of Equation (6) by replacing the probability mass p_l^i of the ith feature with its sample mean. The specific definition is as follows:

$$\mathrm{S}^i = \left\{ \Phi^{-1}\left(\frac{\sum_{t=1}^{|B|} \mathbb{1}(\mathbf{x}_t[i] \le l)}{|B|+1} \right), l \in |k-1| \right\}, \tag{7}$$

where $\mathbf{x}_t[i]$ represents the ith (discrete) feature of the tth input. To estimate the correlation matrix Σ, we employ an online EM method that operates within the buffer B.

Specifically, our objective is to maximize the likelihood that the observed entries (referred to as $\mathbf{X}_O$) of the buffered matrix $\mathbf{X}_B \in \mathbb{R}^{|\mathcal{U}_t| \times |B|}$ can be precisely reconstructed by computing the conditional expectation of Σ. To clarify the notation, we represent $\Sigma^{(t-1)}$ as the empirical correlation acquired in the previous round and $\hat{\Sigma}$ as the target to be approximated in the current round. The log-likelihood function is formulated as follows:

$$\begin{aligned} Q\big(\hat{\Sigma}; \Sigma^{(t-1)}, \mathbf{X}_O\big) &:= \frac{1}{|B|} \sum_{t=1}^{|B|} \mathbb{E}\left[\mathcal{L}(\hat{\Sigma}; \mathbf{x}_t, \mathbf{z}_t) \mid \mathbf{z}_t, \Sigma^{(t-1)} \right] \\ &= \mathrm{const} - \frac{1}{2} \log\det(\hat{\Sigma}) - \frac{1}{2} \mathrm{Tr}\left(\hat{\Sigma}^{-1} G(\Sigma^{(t-1)}, \mathbf{x}_t) \right), \end{aligned} \tag{8}$$

with $\Sigma^{(0)}$ initialized as an identity matrix, and const is a universal constant. $G(\cdot,\cdot)$ aims to map the observed data back to its corresponding latent variables in the normal distribution. Two steps iterate in an alternative fashion to maximize Equation (8) as follows.

E-step. We calculate the empirical $\mathbf{z}_t$ by computing the conditional expectation given $\mathbf{x}_t$ and $\Sigma^{(t-1)}$ using Equation (5). This allows us to express the likelihood $Q(\hat{\Sigma}; \Sigma^{(t-1)}, \mathbf{X}_O)$ in terms of $\hat{\Sigma}$ by replacing $G(\Sigma^{(t-1)}, \mathbf{x}_t) = \mathbb{E}_{t \in B}[\mathbf{z}_t \mathbf{z}_t^\top \mid \mathbf{x}_t, \Sigma^{(t-1)}]$ in Equation (8).

M-step. We solve for $\tilde{\Sigma}$ as $\tilde{\Sigma} = \arg\max_\Sigma Q(\Sigma; \Sigma^{(t-1)}, \mathbf{X}_O)$, a step that is guaranteed to increase the likelihood according to the EM theory (Chapter 3 of [35]). Then, inspired by the method in [5], we replace the current round's correlation with the harmonic mean of the correlation obtained in the last round, denoted as $\Sigma^{(t-1)}$, and $\tilde{\Sigma}$. The objective of this approach is to generate a smooth sequence $\Sigma^{(1)}, \ldots, \Sigma^{(T)}$. This sequence represents a series of local maximizations, unconstrained and monotonically converging. To ensure that empirical maximization fits a normal covariance, we approximate it as follows:

$$\hat{\Sigma} = P_{\mathcal{E}}\big((1-\gamma_t) \Sigma^{t-1} + \gamma_t \tilde{\Sigma} \big), \tag{9}$$

with γ_t is a decaying step size within the range of $(0, 1]$. Additionally, $P_{\mathcal{E}}(\cdot)$ normalizes the correlation as $\mathbf{D}^{-1/2}\hat{\Sigma}\mathbf{D}^{-1/2}$, where $\mathbf{D} = \text{diag}(\hat{\Sigma})$ [16, 62].

4.2 Learning Concept Drift via Adaptive Windowing

The impact of concept drift on online learning algorithms can be significant. Without proper adaptation mechanisms, online learning algorithms may fail to accurately capture the changing patterns in the data and make incorrect predictions. This can lead to degraded performance, reduced accuracy, and increased model bias over time. As discussed and demonstrated in Challenge III (Figure 1(c)) of Section 3.1, the drift of unpredictable classifications can have disastrous effects on online classifiers.

Inspired by the Adaptive Window [3] approach, we propose using a reserved sliding window $\mathcal{S}$ to contain the most recent instances. Upon observing substantial differences in the averages of two sufficiently large sub-windows of $\mathcal{S}$, we conclude that the associated expectations diverge and proceed to eliminate the older portion of the window. A central aspect of the algorithm lies in the definition of the cut and the tests employed. Let h denote the size of $\mathcal{S}$ and h_0 and h_1 denote the sizes of the corresponding sub-windows $\mathcal{S}_0$ and $\mathcal{S}_1$, respectively, such that $h = h_0 + h_1$. The averages of the values within $\mathcal{S}_0$ and $\mathcal{S}_1$ are denoted as $\hat{\mathcal{S}}_0$ and $\hat{\mathcal{S}}_1$, respectively. The cut is defined as follows:

$$\varphi_{\text{cut}} = \sqrt{\frac{1}{20m} \cdot \ln \frac{4h}{\vartheta}}, \tag{10}$$

where $m = \frac{2}{1/h_0 + 1/h_1}$ and $\vartheta \in (0, 1)$ are user-defined confidence parameters, with empirical values set in the range of 0.2–0.5. Upon detection of a drift in the data, $|\hat{\mathcal{S}}_0 - \hat{\mathcal{S}}_1| \geq \varphi_{\text{cut}}$. At this stage, $\mathcal{S}$ slides anew, and the oldest portion is gradually eliminated until: $|\hat{\mathcal{S}}_0 - \hat{\mathcal{S}}_1| < \varphi_{\text{cut}}$.

When concept drift of the data stream is detected, the mechanism of narrowing the window to discard old data points and retaining only the newest data that better represents the current concept is employed. This mechanism ensures that the model is always learning from the most recent and relevant data, thereby quickly adapting to new data distributions.

4.3 Learning Data Geometrics via Adaptive Density Peaks

The transition from observable mixed data to latent space is characterized by the fact that the acquired data instances share a consistent distribution space, and distances between data points are measurable. Consequently, we can explore the latent geometric structure within the latent space, with a focus on propagating limited supervisory information, such as scarce labeled instances, to their structural neighbors.

In this work, we approximate the fundamental geometric characteristics of the data using a clustering structure. To meet the requirements of online learning, we employ a non-iterative, **Density-Peak-based Clustering (DPC)** method [40], whose main process is illustrated in Figure 3. Specifically, we use two metrics to describe the features of each arriving instance $\mathbf{x}_t$, namely local density ρ_t and distance delta δ_t, defined as follows:

$$\rho_t = \sum_{\mathbf{x}_i \in B, i \neq t} e^{-\left(d(\mathbf{x}_t, \mathbf{x}_i)/d_{\text{cut}}\right)^2}, \tag{11}$$

$$\delta_t = \begin{cases} \min_{i:\rho_t < \rho_i}(d(\mathbf{x}_t, \mathbf{x}_i)), & \text{others} \\ \max_i(d(\mathbf{x}_t, \mathbf{x}_i)), \forall i, \rho_t \geq \rho_i \end{cases}, \tag{12}$$

where we employ distance metrics to capture relationships in the reconstructed universal feature space $\mathcal{U}_t$. $d(\mathbf{x}_t, \mathbf{x}_i)$ measures the Euclidean distance between $\mathbf{x}_t$ and $\mathbf{x}_i$, while d_{cut} is a dynamically

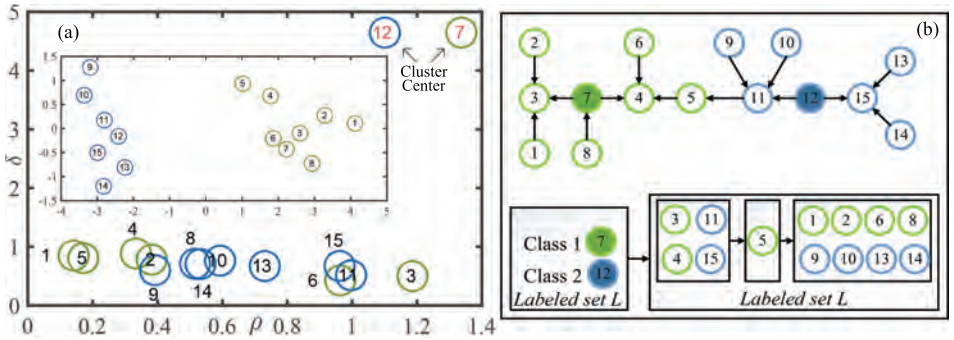


Fig. 3. Illustrating learning data geometrics with 15 instances. (a) The original data distribution and the decision graph. (b) The learned geometric structure and its self-training processes.

adjusted cutoff distance. We also compute the distance δ_t, which reflects the dissimilarity between $\mathbf{x}_t$ and any other instance $\mathbf{x}_i$ with a local density greater than ρ_t. For the cutoff distance d_{cut}, we adopt the formulation $d_{cut} = \lfloor P_{Arr} \times |B| \times (|B| - 1)/2 \rfloor$. The parameter P_{Arr} is empirically set to fall within the range of 1% to 2% based on prior work [49].

We can intuitively determine the cluster centers by considering instances with high ρ_t (indicating a dense neighborhood) and high δ_t (indicating being far from other potential center candidates) as center points. Figure 3(a) illustrates the decision graph for selecting center points based on the harmonic mean of ρ_t and δ_t. Empirical and theoretical evidence [40] suggests that centers chosen based on density-peak criteria and their corresponding clustering are on par with iterative clustering methods like online k-means [8, 20], but with higher computational efficiency, making them more suitable for online learning environments.

After finding the cluster centroids, we create a directed graph by having each $\mathbf{x}_t$ point to its nearest instance $\mathbf{x}_i$ with a higher ρ_t (Figure 3(b)). To leverage limited supervision information, we conduct self-training on this structure. The key is to iteratively select unlabeled instances that can be predicted correctly with high confidence, which are naturally identified as those pointed to by labeled instances. If no labeled instance is pointing to a particular unlabeled instance, we perform a depth-first search to find a label. To handle streaming inputs, our online learner operates as follows: upon arrival of an $\mathbf{x}_t$, we add it to a buffer B and select the most confidently predicted unlabeled instance, $\mathbf{x}_i$, based on the constructed geometric shape. This ensures a prediction delay of at most B, which is more efficient than prior semi-supervised online learners that predict the oldest instance in B with a delay of B timesteps. However, in the most unfavorable scenario, where $\mathbf{x}_t$ cannot be reliably predicted, our method, such as previous research [26], experiences prediction delays. In more favorable situations, $\mathbf{x}_t$ is indicated by already labeled instances, allowing for prediction in the same or near rounds.

Adaptive Truncation Distance to Improve Efficiency. The density-peak clustering algorithm exhibits some limitations, where the choice of the truncation distance, d_{cut}, directly influences the algorithm's clustering efficacy. Empirically selecting the d_{cut} involves choosing the average number of neighbors per data point to be approximately 1–2% of the total number of data points. Furthermore, the DPC algorithm requires the selection of appropriate clustering centers by identifying points with high local densities δ_t and large relative distances d_i. However, selecting suitable clustering centers for complex decision diagrams can be challenging. To address these issues, we propose redefining the local density by utilizing a clustering concept of the nearest neighbor to attenuate the effect of the global truncation distance, d_{cut}, on the local clustering efficacy. Specifically, we

redefine the local density in Equation (11) as follows:

$$\rho(\mathbf{x}_t) = \frac{np(\mathbf{x}_t)}{\sum_{\mathbf{x}_i \in N(\mathbf{x}_t)} d(\mathbf{x}_t, \mathbf{x}_i)}, \tag{13}$$

where $np(\cdot)$ is the number of neighbors of each point, and $N(\mathbf{x}_t)$ is the set of neighbors of point $\mathbf{x}_t$. Specifically, two data points are each other's neighbors when they are K-nearest neighbors to each other. When an initial density peak is discovered, it is assimilated with its lower-density neighbors to form a cluster, denoted as $\forall \mathbf{x}_i \in N(\mathbf{x}_t), \rho(\mathbf{x}_t) > \rho(\mathbf{x}_i)$. Once all the initial density peaks have been surveyed, comparable clusters are combined, denoted as $\mathbf{x}_i \in N(\mathbf{x}_t), \rho(\mathbf{x}_t) < \rho(\mathbf{x}_i)$.

Hence, the employment of the neighborhood approach to assessing the local density of data points without any parameters circumvents the issue of sensitivity to parameters. Notably, since the neighborhood technique precisely mirrors the attribute features of the data points, the local density attained through its application more accurately represents the density of each data point, ultimately leading to an improved clustering effect.

4.4 Online Aggregation for Expedited Convergence

Thus far, we've discussed how the GC model performs reconstruction in the observable feature space and builds a classifier for real-time predictions using the buffer B. We define this classifier as f_O, where "O" signifies that it's trained on the observable mixed features (original mix-typed). We define $y_O = \langle f_O, \mathbf{x}_t^{\text{rec}} \rangle$ as the prediction result. The latent continuous space learned by GC is defined as $y_O = \langle f_O, \mathbf{x}_t^{\text{rec}} \rangle$. However, since the latent space contains only continuous (normal) variables, it cannot fully incorporate the rich information into the model. Certainly, this latent space can expedite the convergence process. However, this involves a tradeoff because if any newly emerging features lack sufficient instances to describe them, it might result in inaccurate latent representations. Our goal is to train an online learner that can harness the advantages of the continuity in the learned latent space without being affected by its inaccuracy. To achieve this, we employ an online aggregate learning approach, wherein two base classifiers, one trained on the original feature space and the other on the latent feature space, work together to provide more accurate predictions.

Let y_Z represent the prediction on the latent representation of $\mathbf{x}_t$, denoted as $\langle f_Z, \mathbf{z}_t \rangle$. The aggregated prediction is given by $\hat{y}_t = \alpha_1 \cdot y_O + \alpha_2 \cdot y_Z$, with $\alpha_1 + \alpha_2 = 1$. These values, α_1 and α_2, determine the importance of the base classifiers f_O and f_Z, respectively. Let $R_O(T)$ and $R_Z(T)$ denote the cumulative risks suffered by f_O and f_Z over T rounds, respectively, calculated as $\sum_{t=1}^{T} \pi(t) \cdot \ell(y_t, y_O)$ and $\sum_{t=1}^{T} \pi(t) \cdot \ell(y_t, y_Z)$. Then, at round $T+1$, α_1 is updated based on risk exponentials as follows [6, 17]:

$$\alpha_1 = e^{-\mu R_O(T)} / (e^{-\mu R_O(T)} + e^{-\mu R_Z(T)}), \tag{14}$$

where $\mu = 2\sqrt{2 \ln 2/T}$ is a tuned parameter. The aggregation of two base classifiers lends our OL-MDISF learner a nice property.

4.5 Algorithm and Complexity Analysis

Based on the above inferences, we design the algorithm of OL-MDISF as in Algorithm 1. Algorithm 1 presents the main steps of our proposed OL-MDISF approach, in which the LDP function echoes to the "Learning Data Geometrics via Local Density-Peaks" module. Below, we derive a step-by-step computational complexity of Algorithm 1.

Algorithm 1: The OL-MDISF Algorithm

Initialize : Classifiers f_O and f_Z, correlation Σ, ensemble factor $\alpha_1 = 0.5$, and cumulative risks $R_O(T) = R_Z(T) = 0$.

Parameters : Buffer B, sparsity c, and endpoint ε.

for $t = |B|, \ldots, T$ **do**
  Receive a mixed data instance $\mathbf{x}_t = (\mathbf{x}_C, \mathbf{x}_D)$;
  Join $\mathbf{x}_t$ in B and establish $\mathrm{GC}(\mathbf{f}, \Sigma)$;
  Estimate $\mathbf{f}$ for continuous $\mathbf{x}_C$ and discrete $\mathbf{x}_D$ with **repeat**
    /* Esitimate $\hat{\Sigma}$ with EM */
    for $t = 1, \ldots, B$ **do**
      E-step: Replace $G(\Sigma^{(t-1)}, \mathbf{x}_t)$ in Eq. (8) with $\mathbf{z}_t = (\mathbf{f}^{-1}(\mathbf{x}_C), \mathrm{cutoff}^{-1}(\mathbf{x}_D), \hat{\mathbf{z}}_M)$ calculated via Eqs. (4) and (5);
      M-step: Obtain $\hat{\Sigma}$ by solving Eqs. (8) and (9);
  until convergence or $\|\hat{\Sigma} - \Sigma^{(t-1)}\|_{\mathrm{Forb}} \leq \varepsilon$;
  /* Predict the oldest input in B */
  Pop vector $\mathbf{x}_{t-|B|+1}$ and reconstruct its latent $\mathbf{z}_{t-|B|+1} = \left(\mathbf{f}^{-1}(\mathbf{x}_C), \mathrm{cutoff}^{-1}(\mathbf{x}_D), \hat{\mathbf{z}}_M\right)$;
  /* Adaptive sliding window */
  Detection of concept drift points: $\varphi_{\mathrm{cut}} = \sqrt{\frac{1}{2m} \cdot \ln \frac{4n}{\vartheta}}$ via Eq. (10);
  /* Adaptive density-peak cluster */
  $f_O = \mathrm{LDP}(\mathbf{x}_t^{rec}, y_t, f_O, d_{cut})$;
  $f_Z = \mathrm{LDP}(\mathbf{z}_t, y_t, f_Z, d_{cut})$;
  Predict the label as $\mathrm{sign}(\hat{y}_{t-|B|+1})$ using $\hat{y}_t = \alpha_1 \cdot y_O + \alpha_2 \cdot y_Z$;
  Reveal the true label $y_{t-|B|+1}$;
  Suffer risks and accumulate $R_O(T)$ and $R_Z(T)$;
  Reweigh coefficient α_1 using Eq. (14);

Function `Local Density-Peaks(LDP)`$(\mathbf{x}_t, y_t, f_b, d_{cut})$**:**
  Calculate δ_i for each sample $\mathbf{x}_i$ using Eq. (12) ;
  Calculate ρ_i for each sample $\mathbf{x}_i$ using Eq. (13);
  Constructing the structure of data space x;
  while *Next (L)* == U **do**
    Training Classifier f_b;
    Update $L \leftarrow L \cup U$;
    Update $U \leftarrow U - L$;
  while *Previous (L)* == U **do**
    Training Classifier f_b;
    Update $L \leftarrow L \cup U$;
    Update $U \leftarrow U - L$;
  return Classifier f_b
End Function;

4.5.1 Time Complexity. The core of the analysis of time complexity consists of six major steps:

Step 1: According to the buffers' size, the data stream's input is divided into multiple buffers, i.e., the number of online model updates.

Steps 2 and 3: In the tth round, we receive an instance $\mathbf{x}_t$ and add it to the set of size $|B|$, establishing the *GC* structure. The establishment process can be completed efficiently in linear time.

Steps 4–8: Evaluate the construction process of g, stop the iteration when convergence or $\|\hat{\Sigma} - \Sigma^{(t-1)}\|_{\text{Forb}} \leq \varepsilon$ is reached and set the required number of iterations to $O(\kappa)$. During the iteration process, finish the online EM operation under the buffer $|B|$. The process of evaluating g takes $O(t * \kappa * |B|)$.

Steps 9 and 10: After obtaining the reconstruction space of $\mathbf{x}_t^{rec}$, the potential distribution space $\mathbf{z}$ can be further obtained. The adaptive window combines historical data with the latest data to detect drift points. The entire setup process can be finished in a linear time.

Steps 11 and 12: After obtaining the $\mathbf{x}$ reconstruction space and the corresponding $\mathbf{z}$ latent space, perform model learning in different spaces. Two rounds of data space construction (the next and previous steps) are performed under the buffer $|B|$, corresponding to steps 21–24 and steps 25–28 in LDP function. The time required for feature space construction is $2 * O(|B|)$. The total time spent is $2 * 2 * O(t * |B|)$.

Steps 13–16: Make the prediction, reveal the true label, calculate risks, and reweigh the coefficient, all after the GC construction and LDP, which can all be done in linear time.

To summarize, our OL-MDISF overall time complexity is the sum of the complexities of the inner and outer loops, which is $O(\kappa \times t \times |B| + T)$.

4.5.2 Spatial Complexity. Considering that the feature space of $\mathbf{x}_t$ and the mix-typed feature space can be completely disjoint at each iteration, we do not apply simplification for the time complexity computing, and the worst runtime of OL-MDISF is finished in $O(t * \kappa * |B|)$. As the number of iterations (i.e., κ) and the maximal dimension of $\mathcal{U}_t$ are under control, the complexity of OL-MDISF is indeed dominated by the length of the input sequence T in practice. Therefore, OL-MDISF is as efficient as other online learning algorithms.

5 Theoretical Analysis and Proofs

In this section, we analyze the theoretical properties of the proposed OL-MDISF approach and provide comprehensive proofs for Lemma and Theorem.

5.1 GC Learning to Latent Representations

The true g in GC [10] is unbiasedly approximated by the monotone function $\hat{g}$ learned from offline data. However, the estimation in our online environment could be skewed inside a buffer B that is proportional to the total data stream. *How much of a difference there is between $\hat{g}$ and g* is still uncertain. This gap is bound by the following two lemmas.

Lemma 5.1 (Continuous case). *Every variable in a random vector $\mathbf{x} \in \mathbb{R}^d$ that follows $\mathbf{x} \sim \text{GC}(\mathbf{z}, \mathbf{f}, \Sigma)$ and has CDF $F(x)$ fulfills $x_i \equiv \hat{g}_i(z_i)$, where $g_i = F_i^{-1} \circ \Phi$. Let $m = \min_{j \in |B|} x_i^{(j)}$ and $M = \max_{j \in |B|} x_i^{(j)}$ stand in for the ith observed the variable's smallest and biggest values in buffer B, respectively. Equation (6) is satisfied by the strictly monotone function $\hat{g}_i$,*

$$\mathrm{P}\left(\sup_{m \leq x \leq M} \left|\hat{g}_i^{-1}(x) - g_i^{-1}(x)\right| > \epsilon\right) \leq 2e^{-c_1 \epsilon^2 |B|}, \tag{15}$$

with ϵ taking an arbitrary value in $(a_1|B|^{-1}, b_1)$ and a_1, b_1, c_1 being constants associated with $F(m)$ and $F(M)$.

Proof. The Dvoretzky-Kiefer-Wolfstein inequality, which constrains the disparity between an empirical CDF and the true CDF, efficiently yields the necessary outcome. we have

$$|B|^{-1} < \epsilon < K_1 \equiv \min\left\{F(m)/4, (1 + F(M))/4\right\}, \tag{16}$$

leading to the definition of $r_c = \frac{|B|}{|B|+1}\mathcal{F}_{|B|}(x)$ in order to achieve $r_c \in \left[F(m)/2, (1 + F(M))/2\right]$. As a result, $\sup_{x\in[m,M]} \left|\Phi^{-1}(r_c) - \Phi^{-1}(F(x))\right| < 2\epsilon \cdot \sup_{r\in[F(m)/2,(1+F(M))/2]} \left|\left(\Phi^{-1}(r)\right)'\right| = K_2 \equiv 1/\min\left\{\phi\left(\Phi^{-1}\left(F(m)/2\right)\right), \phi\left(\Phi^{-1}\left((F(M)+1)/2\right)\right)\right\}$ where $\left(\Phi^{-1}(r)\right)' = 1/\phi\left(\Phi^{-1}(r)\right)$, ϕ and Φ are the PDF and CDF of a standard normal. The proof is completed by adjusting the constants for $2K_2|B|^{-1} < \epsilon < 2K_1K_2$ and the definition of $\hat{g}_i^{-1}$ in Equation (6). □

Lemma 5.2 (Discrete case). *For a random vector x_i in the range $|k|$ with a probability mass function $\{p_l\}_{l=1}^k$, and a normal random variable z_i in $\mathbb{R}^d$ that satisfies $g_i(z) := cuttoff(z_i; \mathrm{S}) = x_i$. The empirical cutoff in Equation (7) holds*

$$\mathrm{P}\left(\|\hat{\mathrm{S}}_i - \mathrm{S}_i\|_1 > \epsilon\right) \leq 2^{|k|} e^{-c_2|B|\epsilon^2/(|k|-1)^2}, \tag{17}$$

with ϵ taking arbitrarily in $\left((|k|-1)a_2|B|^{-1}, (k-1)b_2\right)$ and $a_2, b_2, c_2 > 0$ being constants associated with mass $\{p_l\}_{l=1}^k$.

Proof. The proof is carried out in three steps.

Step 1. We define $s_l^* = \Phi^{-1}\left(\sum_{i=1}^B \mathbb{1}\left(x^i \leq l\right)/|B|\right)$ for $l \in |k-1|$, it is verified that $s_0^* = -\infty$ and $s_k^* = +\infty$, and $\Delta_l^* = \Phi(s_l^*) - \Phi(s_{l-1}^*) = \sum_{i=1}^{|B|} \mathbb{1}(x^i = l)/|B|$. Note that the sequence $|B|\Delta_1^*, \ldots, |B|\Delta_{|k|}^*$ is multinomially distributed with parameters B and $p_1, \ldots, p_k$. We borrow the Bretagnolle-Huber-Carol inequality [46] to have that, for any $\varepsilon > 0$, $\sum_{l=1}^{|k|} \left|\Delta_l^* - p_l\right| < \epsilon$ with probability at least $1 - 2^{|k|}e^{-\frac{1}{2}|B|\epsilon^2}$.

Step 2. For each $l \in |k|$, $\left|\Phi\left(s_l^*\right) - \Phi\left(s_l\right)\right| \leq \sum_{t=1}^{|k|} \left|\Delta_t^* - p_t\right| < \epsilon$. Take $\epsilon > |B|^{-1}$, we arrive at $\Phi\left(s_l\right) - 2\epsilon < \Phi\left(s_l^*\right) \cdot \left(|B|/(|B|+1)\right) = \sum_{i=1}^{|B|} \mathbb{1}(x^i \leq l)/(|B|+1) < \Phi\left(s_l\right) + 2\epsilon$.

Step 3. When $l \in |k-1|$, we have $p_1 \leq \Phi(s_l) \leq \sum_{t=1}^{k-1} p_t$, and by letting $\epsilon < K_1 \equiv \min\{p_1/4, p_k/4\}$, we have $p_1/2 \leq \Phi(s_l^*) \cdot \left(|B|/(|B|+1)\right) \leq 1 - p_k/2$. Therefore, $\|\hat{S}_i - S_i\|_1 = \sum_{l=1}^{k-1} |\hat{s}_l - s_l| \leq 2(k-1)\epsilon/K_2$, where $K_2 = 1/\min\left\{\phi\left(\Phi^{-1}\left(\frac{p_1}{2}\right)\right), \phi\left(\Phi^{-1}\left(1 - \frac{p_k}{2}\right)\right)\right\}$. Adjusting the constants yields $\mathrm{P}\left(\|\hat{S}_i - S_i\|_1 > \epsilon\right) \leq 2\exp\left\{-\frac{1}{8K_2^2} \cdot \frac{|B|\epsilon^2}{(|k|-1)^2}\right\}$, which completes the proof. □

Remark 1. The gap is limited by the observed domain in the buffer B, as per Lemma 5.1, and the empirical $\hat{g}$ converges to the true g in the supremum norm. Lemma 5.2 states that for each discrete variable, the cutoff estimator $\hat{S}_i$ approximates S_i. The empirical estimates of the genuine mapping function, which is cutoff for ordinal features and g for continuous features, approximate it more closely the greater the buffer size $|B|$. We might select a large B to enable an unbiased estimation of the mapping functions for learning effective latent representations in an online environment where new examples are constantly arriving.

5.2 Online Aggregation Convergence

We derive the regret bound of our OL-MDISF learner.

Theorem 5.3. *Over T rounds, we have*

$$\sum_{t=1}^{T} \pi(t) \cdot \ell(y_t, \hat{y}_t) \leq \min\{R_O(T), R_Z(T)\} + \sqrt{2T \ln 2}. \tag{18}$$

Proof. The proof is completed with four observations.

Observation 1. *We define a quantitative small*

$$Q_T = \exp(-\mu R_O(T)) + \exp(-\mu R_Z(T)), \tag{19}$$

and it is verified that $Q_1 = 2$ and $\ln(Q_T/Q_1) = -\mu \min\{R_O(T), R_Z(T)\} - \ln 2$.

Observation 2. *By expanding small*

$$R_p(T) = R_p(T-1) + r_p(T), \quad p \in \{O, Z\}, \tag{20}$$

where $r_p(T)$ is the instantaneous risk suffered by y_O or y_Z at the Tth round, we arrive at $\ln(Q_T/Q_{T-1}) = \ln[\alpha_1 \exp(-\mu r_O(T)) + \alpha_2 \exp(-\mu r_Z(T))]$, with α_1 defined in Equation (14).

Observation 3. *We leverage the convexity of loss function and adapt the Hoeffding Inequality (cf. Appendix A.1.1 in [6]) to deduce $\ln[\alpha_1 \exp(-\mu r_O(T)) + \alpha_2 \exp(-\mu r_Z(T))] \leq -\mu R_T + \mu^2/8$.*

Observation 4. *Over T rounds we have $\ln(Q_T/Q_{T-1}) + \cdots + \ln(Q_2/Q_1) = \ln(Q_T/Q_1) \leq -\mu \sum_{t=1}^{T} R_t + T \cdot (\mu^2/8)$. Collecting the above four observations, we have $\sum_{t=1}^{T} R_t \leq \min\{R_O(T), R_Z(T)\} + (\mu/8)T + \ln 2/\mu$, in which plugging $\mu = 2\sqrt{2 \ln 2/T}$ completes the proof.* □

Remark 2. It is immediate that $\lim_{T\to\infty}(\sqrt{2T \ln 2}/T) = 0$, which means that our OL-MDISF is asymptotically no-regret compared to the winner of the two base predictors, whichever yields a thus-far better online classifier based on how well the underlying geometric structures have been learned. We also empirically show in the next section that both the original, mix-typed features and the latent normals can convey discriminant information, such that either base classifier may prevail yet is unforeseeable, necessitating the ensembling which reap their merits simultaneously to better prediction.

6 Experiments

This section presents empirical evidence to validate the effectiveness of our proposed OL-MDISF algorithm. The experiments aim to address the following **Research Questions (RQs)**:

— *RQ1*. Does our OL-MDISF outperform the state-of-the-art models?
— *RQ2*. How can GC effectively handle mix-typed features in an online setting?
— *RQ3*. How does concept drift dataflow affect model performance?
— *RQ4*. Can density-peak profile the geometric structure of data?
— *RQ5*. Does online ensembling yield better accuracy?

6.1 Experimental Settings

(1) *Datasets*. We have evaluated 14 normal datasets and six drift datasets in our study. Out of these, 13 of the 14 normal datasets were obtained from the UCI[1] [9] database, while the remaining one was sourced from **Massive Online Analysis (MOA)** [4] to simulate a realistic streaming setup. Furthermore, we used four of the drift datasets from the real world, and

[1] https://archive.ics.uci.edu/.

Table 2. Statistics of the Mix-typed Datasets

Dataset	#Instance	#Feature	Dataset	#Instance	#Feature
wpbc	198	33	dna	949	180
ionosphere	351	34	german	1,000	24
wdbc	569	30	splice	3,190	60
australian	690	14	kr-vs-kp	3,196	36
credit-a	690	15	magic04	19,020	10
wbc	699	9	a8a	22,696	123
diabetes	768	8	stream	10,000	1,000

Table 3. Statistics of the Drifted Datasets

Dataset	#Instance	#Attr.	#Chunk	#Concept Change
Agrawal Mixed	80,000	9	1,000	1-2-3-4-1-2-3-4
SEA Abrupt	80,000	3	1,000	1-2-3-4-1-2-3-4
Electricity	45,312	8	1,000	1-2-3-4-1-2-3-4
Hyperplane Abrupt	80,000	3	1,000	1-2-3-4-1-2-3-4
MOA Abrupt	10,000	5	5,000	1-2
MOA Gradual	10,000	100	5,000	1-2

two were generated by MOA [4] to simulate a realistic streaming drift setup. Tables 2 and 3 summarize their statistics.

(2) *Evaluation Protocol*. We evaluated the algorithm performance based on the CER, a metric referenced from prior research [17, 57]. CER measures the ratio of incorrect predictions to all observed instances, providing a standard for performance assessment. The operation form is:

$$\mathrm{CER} = 1/t \sum_{j<t} \langle y_i \neq \mathrm{sign}\,(\tilde{y}_i) \rangle\,, \tag{21}$$

where $\langle \cdot \rangle$ takes the value 1 if the argument is true, and 0 otherwise.

(3) *Compared Methods*. For this comparative study, eight cutting-edge rivals designed for processing online mix-typed streaming characteristics are used, with their major ideas described below:

— *FOBOS* [45]: It sets up a baseline so that online learners can directly train on the features that have been seen. To make it work in a variable feature space, we fill in the blanks with zeros. The goal of the projected subgradient method is to get a sparse solution so that redundant features with (almost) zero coefficients can be cut off.

— *OMR* [14]: It is tailored to handle the missing labels of data streams. Its key idea is to learn the stream shape of the data stream, characterize the high-dimensional features with low-dimensional features, and try to preserve the high-dimensional features.

— *OLSF* [57]: Its emergence aims to address the challenge of learning in increasingly large feature spaces. The core idea is to employ a combination of strategic and passive learning strategies, where weight coefficients are assigned to new features. By monitoring whether new features contribute to the decision boundary, weight updates for new features are conducted in conjunction with existing features.

— *OVFM* [16]: It is tailored to handle mixed data streams. Its key idea is to go through Gaussian cointegration modeling to establish the association of observable data with

potentially distributed data, which can effectively eliminate mixed data from affecting the convergence of the model.

— *OSLMF* [50]: It is exploring the challenges and solutions in online semi-supervised learning, especially for the case of having a heterogeneous feature space. The article proposes a semi-supervised online learner that can handle mixed types of features.

— *OLI^{2}DS* [53]: It is designed to address the challenge of learning from data streams characterized by incomplete feature spaces and dynamically imbalanced class distributions. The algorithm integrates the principles of empirical risk minimization and an F-measure optimization strategy, enabling it to effectively capture incomplete feature space data and adapt to the skewed class distribution.

— *OLCDS* [64]. It is an online learning model designed to handle dynamic feature spaces in data streams through shared and new feature spaces. It enables consistent learning by linking old and new features, allowing adaptability to evolving data.

— *OLIFL* [54]. It is an online learning model designed to handle incomplete features and labels within data streams. It enables adaptive updates in real-time, ensuring effective performance even when data input is partially missing.

(4) Implementation Details.

OL-MDISF Hyperparameter Settings: We used two types of data stream features: trapezoidal data streams and capricious data streams. Trapezoidal data streams are characterized by the continuous addition of new features over time, leading to an increase in dimensionality. On the other hand, capricious data streams involve unpredictable changes in both new and old features, with new features appearing randomly and old features gradually fading away over time. In both scenarios, we randomly removed 50% of the labels from the datasets to simulate a semi-supervised learning environment. We also conducted additional experiments to investigate label scarcity, introducing random label-missing percentages of 10%, 50%, 70%, and 90%. The initial value of the integration value α_1 for different spaces is 0.5, and the corresponding initial value of α_2 is also 0.5. All the experiments are run on a computer server that has a 3.00 GHz Intel Xeon Gold 6248R with 24 cores, 256 GB RAM, and Tesla V100S-PCIE-32 GB * 1.

Comparison Method Settings: We apply the same settings from prior studies [30, 50]. To ensure a fair comparison, all competing models underwent meticulous hyperparameter tuning across various data stream formats of all datasets to achieve optimal performance. The configurations of trapezoidal data streams and capricious data streams for all datasets are consistent with those used in our OL-MDISF algorithm.

6.2 Performance Comparison (RQ 1)

Based on the analysis of Tables 4 and 5, we applied the Wilcoxon signed-ranks test (p-value) [7] and the Friedman test (F-rank) [7] to assess statistical significance. The results indicate that OL-MDISF achieved the highest accuracy in most cases, outperforming competitors in only 7 out of 196 settings. All p-values were below 0.05, except for OL-MDISF in capricious data streams, supporting OL-MDISF's significantly superior prediction accuracy at a 95% confidence level. According to the F-rank analysis, OVFM and OLSF exhibit similar performance, both ranking below OL-MDISF. OMR follows closely, while FOBOS ranks the lowest. This ranking aligns with the design characteristics of each competing algorithm: OVFM and OLSF are optimized to adapt to dynamic feature spaces, OMR assumes a fixed feature space, and FOBOS lacks specific mechanisms for handling feature space dynamics and label scarcity. OLCDS leverages shared and newly added feature spaces to effectively manage randomness and uncertainty in data streams, while OLIFL addresses the challenges posed by missing features and labels in data streams.

Table 4. Performance Comparison of OL-MDISF Against Baseline Methods on Capricious Data Streams

	Capricious Data Streams							
Dataset	FOBOS	OMR	OVFM	OSLMF	OLI^{2}DS	OLCDS	OLIFL	**OL-MDISF**
wpbc	0.248 ± 0.000•	0.320 ± 0.000	0.309 ± 0.000•	0.567 ± 0.001	0.318 ± 0.000	0.462 ± 0.000	0.384 ± 0.000	0.311 ± 0.000
ionosphere	0.479 ± 0.000	0.418 ± 0.000	0.269 ± 0.000•	0.466 ± 0.000	0.481 ± 0.000	0.384 ± 0.001	0.373 ± 0.000	0.334 ± 0.001
wdbc	0.628 ± 0.000	0.399 ± 0.000	0.179 ± 0.000	0.110 ± 0.000•	0.481 ± 0.000	0.324 ± 0.000	0.396 ± 0.000	0.167 ± 0.001
australian	0.455 ± 0.000	0.492 ± 0.001	0.255 ± 0.000	0.194 ± 0.000	0.199 ± 0.000	0.352 ± 0.000	0.397 ± 0.000	**0.187 ± 0.001**
credit-a	0.445 ± 0.000	0.484 ± 0.000	0.484 ± 0.000	0.416 ± 0.000	0.517 ± 0.000	0.528 ± 0.000	0.463 ± 0.000	**0.402 ± 0.000**
wbc	0.162 ± 0.000	0.461 ± 0.000	0.092 ± 0.000	0.059 ± 0.000•	0.167 ± 0.000	0.299 ± 0.000	0.189 ± 0.000	0.089 ± 0.000
diabetes	0.349 ± 0.000	0.426 ± 0.000	0.399 ± 0.000	0.331 ± 0.004	0.449 ± 0.000	0.462 ± 0.000	0.398 ± 0.000	**0.330 ± 0.000**
dna	0.511 ± 0.000	0.496 ± 0.000	0.282 ± 0.000	0.229 ± 0.000	0.304 ± 0.000	0.419 ± 0.000	0.443 ± 0.000	**0.227 ± 0.001**
german	0.700 ± 0.000	0.372 ± 0.000	0.321 ± 0.000	0.227 ± 0.000	0.345 ± 0.000	0.398 ± 0.000	0.378 ± 0.000	**0.218 ± 0.000**
splice	0.519 ± 0.000	0.420 ± 0.001	0.498 ± 0.000	0.424 ± 0.000	0.497 ± 0.000	0.442 ± 0.000	0.509 ± 0.000	**0.418 ± 0.001**
kr-vs-kp	0.478 ± 0.000	0.242 ± 0.000	0.280 ± 0.000	0.241 ± 0.000	0.331 ± 0.000	0.371 ± 0.000	0.355 ± 0.000	**0.218 ± 0.000**
magic04	0.689 ± 0.000	0.438 ± 0.000	0.317 ± 0.000	0.091 ± 0.000•	0.317 ± 0.000	0.261 ± 0.000	0.306 ± 0.000	0.132 ± 0.000
a8a	0.401 ± 0.003	0.368 ± 0.001	0.191 ± 0.001	0.086 ± 0.001•	0.181 ± 0.001	0.302 ± 0.000	0.203 ± 0.000	0.104 ± 0.001
stream	0.621 ± 0.000	0.471 ± 0.000	0.231 ± 0.000	0.224 ± 0.000	0.452 ± 0.000	0.327 ± 0.000	0.283 ± 0.000	**0.201 ± 0.000**
Loss/win	1/13	0/14	2/14	4/10	0/14	0/14	0/14	**7/91***
p-value	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0002	––
F-rank	6.428	5.892	3.786	2.535	4.928	5.679	5.250	1.501

The experiment was repeated 10 times for each dataset, and the average CER was calculated along with the variance. We collected experimental results for 14 datasets in the context of a trapezoidal data stream, representing OL-MDISF's performance with bullets (•) indicating less favorable outcomes and asterisks (*) showing the total number of wins and losses for OL-MDISF. For each dataset row, the best-performing method (*i.e.*, lowest CER) is shown in **bold**.

Table 5. Performance Comparison of OL-MDISF on Trapezoidal Data Streams

	Trapezoidal Data Streams							
Dataset	FOBOS	OMR	OLSF	OSLMF	OLI^{2}DS	OLCDS	OLIFL	**OL-MDISF**
wpbc	0.237 ± 0.000	0.345 ± 0.000	0.366 ± 0.001	0.235 ± 0.003	0.242 ± 0.000	0.301 ± 0.000	0.223 ± 0.000	**0.197 ± 0.001**
ionosphere	0.342 ± 0.000	0.443 ± 0.000	0.230 ± 0.000	0.225 ± 0.000	0.239 ± 0.318	0.243 ± 0.000	0.281 ± 0.000	**0.183 ± 0.001**
wdbc	0.577 ± 0.000	0.460 ± 0.000	0.347 ± 0.000	0.187 ± 0.000	0.353 ± 0.000	0.231 ± 0.000	0.256 ± 0.000	**0.186 ± 0.001**
australian	0.497 ± 0.000	0.491 ± 0.000	0.486 ± 0.000	0.356 ± 0.000	0.497 ± 0.000	0.491 ± 0.000	0.385 ± 0.000	**0.335 ± 0.001**
credit-a	0.445 ± 0.000	0.415 ± 0.000	0.312 ± 0.000	0.186 ± 0.000	0.517 ± 0.000	0.472 ± 0.000	0.415 ± 0.000	**0.182 ± 0.001**
wbc	0.345 ± 0.000	0.394 ± 0.000	0.455 ± 0.000	0.219 ± 0.000	0.544 ± 0.000	0.298 ± 0.000	0.226 ± 0.000	**0.205 ± 0.001**
diabetes	0.349 ± 0.000	0.376 ± 0.000	0.331 ± 0.000	0.170 ± 0.000	0.449 ± 0.000	0.218 ± 0.000	0.209 ± 0.000	**0.168 ± 0.001**
dna	0.518 ± 0.000	0.496 ± 0.000	0.499 ± 0.000	0.462 ± 0.000	0.305 ± 0.000	0.416 ± 0.000	0.442 ± 0.000	**0.366 ± 0.001**
german	0.300 ± 0.000	0.381 ± 0.000	0.407 ± 0.000	0.227 ± 0.000	0.345 ± 0.000	0.218 ± 0.000	0.273 ± 0.000	**0.209 ± 0.001**
splice	0.500 ± 0.000	0.493 ± 0.000	0.375 ± 0.000	0.311 ± 0.000	0.497 ± 0.000	0.372 ± 0.000	0.354 ± 0.000	**0.306 ± 0.001**
kr-vs-kp	0.482 ± 0.000	0.523 ± 0.000	0.239 ± 0.000	0.221 ± 0.000	0.589 ± 0.000	0.301 ± 0.000	0.227 ± 0.000	**0.218 ± 0.001**
magic04	0.665 ± 0.000	0.529 ± 0.000	0.374 ± 0.000	0.348 ± 0.000	0.423 ± 0.000	0.387 ± 0.000	0.436 ± 0.000	**0.327 ± 0.001**
a8a	0.375 ± 0.000	0.482 ± 0.003	0.273 ± 0.004	0.179 ± 0.001	0.458 ± 0.000	0.271 ± 0.000	0.351 ± 0.000	**0.163 ± 0.001**
stream	0.615 ± 0.000	0.472 ± 0.000	0.233 ± 0.000	0.230 ± 0.000	0.455 ± 0.000	0.261 ± 0.000	0.283 ± 0.000	**0.217 ± 0.001**
Loss/win	0/14	0/14	0/14	0/14	0/14	0/14	0/14	**0/98***
P-value	0.0003	0.0001	0.0003	0.0003	0.0001	0.0003	0.0003	––
F-rank	6.608	6.643	4.928	2.357	6.178	4.250	3.965	1.071

The experiment was repeated 10 times for each dataset, and the average CER was calculated along with the variance. We collected experimental results for 14 datasets in the context of a trapezoidal data stream, representing OL-MDISF's performance with bullets (•) indicating less favorable outcomes and asterisks (*) showing the total number of wins and losses for OL-MDISF. For each dataset row, the best-performing method (*i.e.*, lowest CER) is shown in **bold**.

Overall, this indicates that the OL-MDISF algorithm demonstrates strong adaptability and robustness in handling dynamic and complex data stream environments. Particularly in scenarios involving variability and gradual feature changes, OL-MDISF is able to maintain consistently high performance, not only excelling in individual scenarios but also performing stably across

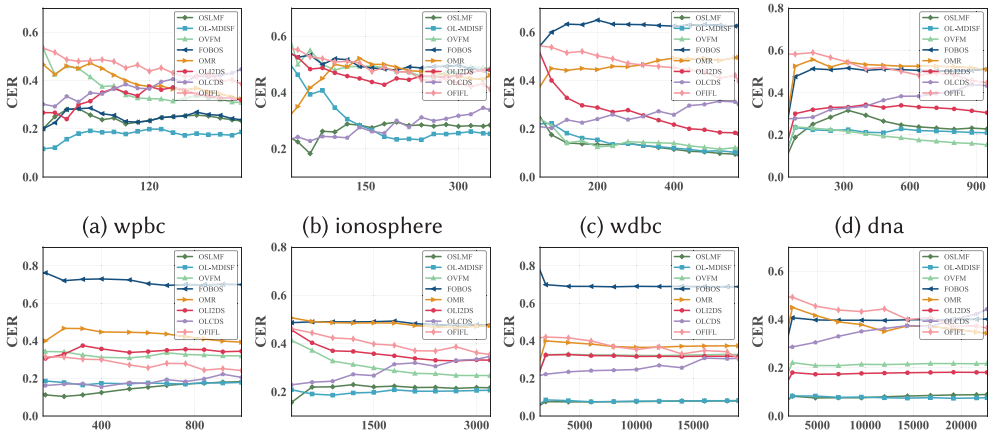


Fig. 4. Three methods' trends in CER within capricious data streams.

diverse conditions. In summary, OL-MDISF outperforms other algorithms in effectively addressing uncertainty and missing features in data streams, showcasing superior generalization ability and applicability in dynamic environments.

6.3 GC Performance (RQ 2)

The comparison between our OL-MDISF and OMR provides the answer. While both utilize the geometric data structure for online semi-supervised learning, OMR does not accommodate inputs with mix-typed streaming features. We observe that OL-MDISF achieves a 48.2% lower CER compared to OMR. Additionally, in Figure 4, OL-MDISF exhibits CER values 0.41% and 76.63% lower than OMR on the kr-vs-kp and a8a datasets, both of which naturally feature mix-typed data. OL-MDISF distinguishes itself in handling mix-typed features in an online setting, with its low CER across datasets underscoring its effectiveness. For instance, in the ionosphere dataset, OL-MDISF maintains a CER around 0.2, significantly lower than competing models like FOBOS and OL^{2}DS, which fluctuate above 0.4. This performance highlights OL-MDISF's ability to manage the complex challenges associated with mix-typed streaming data, showcasing its resilience and adaptability in comparison to other methods.

The superior performance of OL-MDISF stems from three core mechanisms: copula-based latent space modeling, adaptive drift detection, and label proximity alignment. The copula model creates a comprehensive latent space that captures complex relationships between heterogeneous features, providing a stable foundation even with incomplete data. The adaptive sliding window allows OL-MDISF to detect drift points in real-time, ensuring that decision boundaries remain effective as data distributions change. Additionally, by leveraging geometric structural relationships for label proximity, OL-MDISF effectively handles sparse supervision. Together, these components enable OL-MDISF to maintain accuracy and stability in dynamic data streams, making it a robust solution for mix-typed online learning.

6.4 Concept Drift (RQ 3)

The analysis of concept drift detection in classification tasks underscores the critical impact of drift on model performance, particularly in dynamic data environments. When concept drift occurs,

Table 6. Comparison of CER for OL-MDISF and Baselines Across 6 Synthetic Datasets Under Capricious Stream Scenarios (e.g., Abrupt and Gradual Drifts)

	Capricious Data Streams								
Dataset	FOBOS	OMR	OLSF	OVFM	OSLMF	OLI^{2}DS	OLCDS	OLIFL	OL-MDISF
SEA Abrupt	0.301	0.475	0.375	0.366	0.309	0.327	0.382	0.409	**0.283**
Hyperplane Abrupt	0.485	0.462	0.392	0.402	0.372	0.402	0.388	0.478	**0.352**
Agrawal Mixed	0.498	0.416	0.434	0.446	0.398	0.382	0.401	0.392	**0.376**
Electricity	0.393	0.421	0.442	0.482	0.402	0.398	0.523	0.482	**0.383**
MOA Abrupt	0.426	0.384	0.409	0.426	0.398	0.352	0.417	0.529	**0.364**
MOA Gradual	0.497	0.427	0.452	0.392	0.385	0.358	0.355	0.328	**0.326**

Experimental results (CER) for six datasets in the case of capricious data streams. This table reflects adaptability to different stream dynamics. For each dataset row, the best-performing method (i.e., lowest CER) is shown in **bold**.

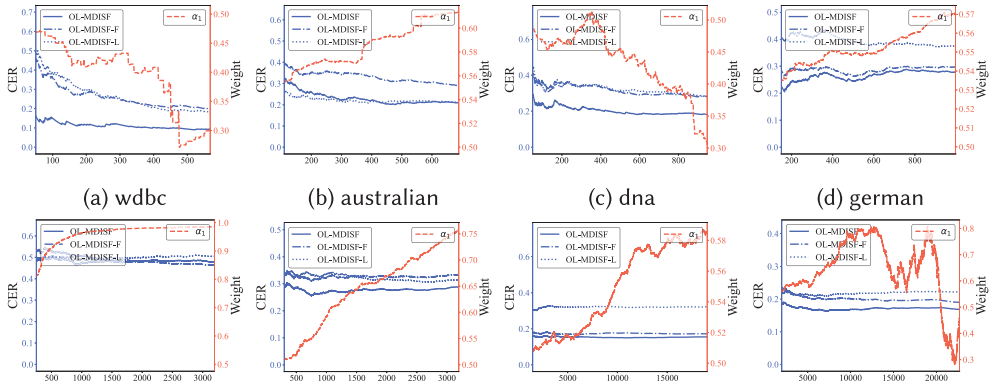
it can lead to a rapid degradation in model accuracy if not promptly addressed. Our approach to mitigating these effects involves dynamically resizing the buffer block, which allows the generative model to update in anticipation of drift events, effectively preventing a collapse in performance.

Empirical results from the SEA Abrupt, MOA Gradual, and MOA Abrupt datasets clearly demonstrate the advantages of this adaptive method. Specifically, our proposed approach achieves CERs that are 15.32%, 8.54%, and 8.41% lower than those of the OSLMF algorithm on the respective datasets (Table 6). These significant reductions highlight the efficacy of our approach in detecting and responding to drift points. By adaptively windowing the drift detection process and adjusting the buffer pool size in real-time, our method sustains high predictive accuracy, ensuring robust classification performance even as underlying data patterns shift. This adaptive capability not only minimizes the adverse impacts of concept drift but also enhances the model's resilience in fluctuating data streams, positioning it as an effective solution for accurate, real-time classification in capricious environments.

6.5 Geometric Structure Learning (RQ 4)

The experimental results in Tables 4 and 5 provide strong evidence that density peaks can indeed profile the geometric structure of data, as demonstrated by the superior performance of our algorithm (OL-MDISF) compared to OVFM. By effectively leveraging density peaks, OL-MDISF captures the intrinsic geometric distribution within the data, allowing it to propagate sparse label information across the dataset with high accuracy. This capability is particularly beneficial in scenarios with limited labeled data, where the algorithm can tap into the wealth of unlabeled data to reinforce learning.

By tracking the entire online learning process (Figure 4), our comparative analysis reveals substantial improvements in CER for OL-MDISF over OVFM, specifically achieving 23.92%, 29.28%, 13.93%, and 54.97% lower CERs on the australian, german, kr-vs-kp, and a8a datasets, respectively. These reductions underscore OL-MDISF's exceptional capacity to interpret and utilize the geometric structure of data for label propagation, thereby enhancing overall prediction accuracy. By capitalizing on density-based geometric profiling, OL-MDISF transforms the distributional characteristics of data into meaningful structural insights, enabling robust and reliable classification even in partially labeled data streams. Our approach not only boosts prediction performance but also establishes a comprehensive understanding of the data landscape, proving the effectiveness of density peaks in representing complex geometric patterns within diverse data streams.



(a) wdbc (b) australian (c) dna (d) german

(e) splice (f) kr-vs-kp (g) magic04 (h) a8a

Fig. 5. Temporal variation of ensemble weight α_1 and CERs of OL-MDISF and its ablation variant OL-MDISF-F and OL-MDISF-L.

6.6 Online Aggregation (RQ 5)

To empirically investigate OL-MDISF's adaptation to the combination of two base classifiers, each focused on different feature spaces, we closely monitored the trends of the integration coefficients α_1, as shown in Figure 5. It's important to note that the sum of the parameters α_2 and α_1 equals 1, and their trend curves exhibit symmetry. While the patterns of α_1 varied across datasets, OL-MDISF's accuracy consistently improved, indicated by decreasing CER values. The values of α_1 play a critical role in determining the relative importance of the classifiers trained in the observed feature space, with higher values indicating greater significance and lower values emphasizing classifiers in the latent normal space. Our aggregation strategy facilitates the adaptive learning of coefficients α_1 and α_2 from streaming inputs, eliminating the need to preselect superior classifiers and reducing associated overhead.

For further investigation, we conducted an ablation experiment to plot CERs for prediction using only features in the observation space and features in the latent space. Then, we compared OL-MDISF with two simplified algorithms, OL-MDISF-F and OL-MDISF-L, respectively. We observe that OL-MDISF-F and OL-MDISF-L are inferior to the ensemble OL-MDISF with a consistently higher CER (thus lower accuracy). These findings suggest the effectiveness of online ensembling and empirically validate the tightness of the bound derived in Theorem 5.3.

6.7 Ablation Study

Buffer Size. By comparing ablation experiments with different buffers, our analysis revealed that the accuracy of the model (indicated by a lower CER) is positively correlated with the size of the buffer, as a larger buffer provides more information regarding the geometric spatial distribution. Moreover, the pseudo-label obtained via geometric spatial clustering is more proximal to the true label when a larger buffer is employed. It is important to note that while a larger buffer leads to more precise geometric clustering modeling, the EM process takes longer to converge. Additionally, our investigation disclosed that the impact of buffer size (B) on CER (Table 8) is more pronounced for smaller datasets, as opposed to larger datasets (e.g., dna, german, splice, kr-vs-kp, magic04, a8a, stream).

Difference Missing Ratio. We evaluated the learning performance of different linear models in the context of the semi-supervised performance of the geometric spatial structure. Given the impact of

Table 7. Effectiveness Analysis of w/o. Fully Observed Data (w/o. F), w/o Latent Data (w/o. L), and Integration of Observed and Latent Data Spaces

	Capricious Data Streams														
	Type	wpbc	ionosphere	wdbc	australian	credit-a	wbc	diabetes	dna	german	splice	kr-vs-kp	magic04	a8a	stream
w/o. L	Trapezoidal	0.293	0.291	0.218	0.354	0.454	0.129	0.380	0.256	0.310	0.489	0.324	0.190	0.228	0.472
w/o. F		0.308	0.365	0.139	0.261	0.438	0.072	0.378	0.300	0.343	0.495	0.294	0.318	0.217	0.460
OL-MDISF		**0.258**	**0.242**	**0.105**	**0.249**	**0.417**	0.073	**0.346**	**0.149**	**0.258**	**0.477**	**0.275**	**0.174**	**0.184**	**0.452**
w/o. L	Capricious	0.222	0.405	0.197	0.290	0.438	0.114	0.443	0.281	0.297	0.464	0.332	0.172	0.189	0.460
w/o. F		0.303	0.259	0.181	0.213	0.454	0.074	0.385	0.279	0.374	0.505	0.314	0.322	0.223	0.460
OL-MDISF		**0.207**	0.336	**0.093**	**0.212**	**0.417**	**0.056**	**0.374**	**0.181**	**0.278**	0.477	**0.288**	**0.155**	**0.168**	**0.441**

For each dataset row, the best-performing method (*i.e.*, lowest CER) is shown in bold.

Table 8. Sensitivity Analysis of OL-MDISF Under Varying Buffer Sizes, Missing Label Ratios, Decaying Step Sizes, and Cutoff Distances

	Capricious Data Streams						
	Dataset	dna	german	splice	kr-vs-kp	magic04	a8a	stream
Buffer Size B	2	0.2539	0.2301	0.2404	0.2518	0.0864	0.1008	0.4469
	10	0.2434	0.1831	0.2561	0.2334	0.0847	0.1007	0.4618
	20	0.2403	0.2191	0.2485	0.2359	0.0791	0.0893	0.4638
	40	0.2508	0.2331	0.2369	0.2346	0.0846	0.0939	0.4582
Miss Label Ratio	10	0.3582	0.3251	0.2717	0.2675	0.1022	0.1339	0.4246
	50	0.2507	0.2281	0.2579	0.2618	0.0886	0.1091	0.4539
	70	0.2239	0.2011	0.2391	0.2205	0.1021	0.0948	0.4599
	90	0.1812	0.2031	0.2153	0.2086	0.1021	0.0844	0.4647
Decaying Step Size γ_t	0.1	0.2244	0.2879	0.3545	0.3203	0.2808	0.2152	0.1972
	0.3	0.2255	0.2838	0.3279	0.3145	0.279	0.2004	0.2076
	0.5	0.2119	0.31	0.3545	0.2926	0.2784	0.2026	0.1958
	0.7	0.2203	0.283	0.3358	0.3067	0.2779	0.2001	0.2048
Cutoff Distance d_{cut} ($\lvert B\rvert = 10$)	$P_{Arr} = 5\%$	0.2487	0.278	0.3225	0.3051	0.2784	0.2031	0.2128
	$P_{Arr} = 10\%$	0.2529	0.281	0.3373	0.3123	0.2784	0.2076	0.209
	$P_{Arr} = 15\%$	0.2161	0.2818	0.3376	0.3069	0.2787	0.1983	0.2136
	$P_{Arr} = 20\%$	0.2255	0.2636	0.3357	0.3529	0.2791	0.1976	0.2048

Results are shown as CER values across 6 datasets, highlighting the model's robustness to memory and label incompleteness. For each dataset row, the best-performing method (*i.e.*, lowest CER) is shown in **bold**.

labeling ratio on classification accuracy, we conducted simulations to evaluate the performance of algorithms in a realistic setting, with label ratios set at 10%, 50%, 70%, and 90% for CER. We observe that the model achieves the same stable model results (CER) in the face of different missing labels from 20 to 80% under capricious and trapezoidal data streams. The semi-supervised basic linear model used for OL-MDISF in this article is SVC, and the best result under different missing label control experiments is the LR model.

Decaying Step Size. Our results indicate that a smaller decaying step size (γ_t) contributes to lower CER, as observed in Table 8. When $\gamma_t = 0.1$, the model achieves better performance across datasets, suggesting more stable updates in the iterative EM process. However, increasing γ_t to 0.7 leads to a noticeable rise in CER, particularly for datasets like splice and kr-vs-kp, where the instability of large step sizes disrupts the pseudo-labeling process. This confirms that moderate step sizes strike a balance between convergence speed and classification accuracy, allowing the model to adapt effectively without sacrificing performance due to excessive fluctuations.

Cutoff Distance. Cutoff distance (d_{cut}) determines the local neighborhood size in geometric clustering, significantly influencing CER. As seen in Table 8, smaller P_{Arr} values (e.g., 5%) correspond

to lower CER, implying that a tighter neighborhood improves pseudo-labeling accuracy. Conversely, increasing P_{Arr} to 20% degrades performance, particularly in magic04 and kr-vs-kp. This suggests that while a larger cutoff distance accelerates processing, it introduces noise in clustering, negatively affecting classification accuracy. Therefore, optimizing d_{cut} is essential for maintaining a balance between computational efficiency and accuracy, ensuring that the model remains both scalable and effective in dynamic data environments.

Data Spatial Aggregation. To investigate the impact of aggregating various data stream spaces on online learning performance, we conducted a comparative analysis across three variants: w/o fully observed data (w/o. F), w/o latent data (w/o. L), and the integration of observed and latent data spaces. Examination of Table 7 reveals that the aggregation of different data spaces consistently yields the best performance across the majority of datasets. This indicates that aggregation effectively merges information from diverse data spaces, thereby enhancing outcomes. Additionally, a comparison between the w/o. L and w/o. F results reveal that the performance when excluding fully observed data is inferior to that when excluding latent data (w/o. L), suggesting that fully observed data contains more substantial spatial information than latent data. Further analysis of the trend α_1 curve in Figure 5 demonstrates that the influence of different data spaces on the model varies and that dynamic, adaptive adjustments can steer the model toward optimal convergence.

7 Conclusions

In this article, we aimed to extend the frontiers of online learning from doubly streaming data. We explore a novel learning problem called OL-MDISF which doesn't make assumptions about feature types or learning labels, outperforming prior studies that impose assumptions on either or both in terms of flexibility and applicability. We use the GC to model correlations between discrete and continuous variables in a latent normal space, mitigating abrupt updates for rapid convergence in handling mix-typed streaming features. To mitigate the impact of data drift, we identify the drift points using adaptive dynamic windows, thereby ensuring that the model is trained on both historical and current data. We reveal the geometric structure of incoming instances using density-peak clustering. This approach allows us to extend labeling information to neighboring instances that are currently unlabeled. A theoretical analysis rationalized the design of our proposed approach, and an empirical study further substantiated its effectiveness and superiority over state-of-the-art competitors.

OL-MDISF assumes that data streams follow an **Independent and Identically Distributed (IID)** setting. Although it considers concept drift, its adaptability to long-term distribution evolution, multimodal data fusion, and non-IID data remains limited. Additionally, its density-peak clustering method may experience degraded label propagation performance when data density is uneven or class imbalance is extreme, particularly in scenarios with highly sparse labels. Furthermore, the method does not fully account for the impact of outliers and noisy data. Given that GC relies on statistical correlation modeling, deviations from its assumptions—such as the presence of extreme outliers or non-Gaussian distributions—may affect overall learning performance.

To enhance the adaptability and generalization capability of OL-MDISF, future research can focus on three key directions: (1) Conducting in-depth investigations into the dynamic characteristics of open feature spaces and exploring more constrained feature evolution mechanisms to improve the model's adaptability in complex streaming data environments. (2) Integrating techniques from multiple domains to enhance the performance and applicability of online learning models. This includes incorporating adaptive parameter optimization and robust statistical modeling to improve resilience against non-IID data and noisy inputs. (3) Addressing challenges in real-world applications, such as mobile malware detection, by further optimizing the model's performance in dynamic environments, ensuring robustness in handling data complexity and uncertainty in practical scenarios.

References

[1] Charu C. Aggarwal. 2007. *Data Streams: Models and Algorithms*. Vol. 31, Springer.

[2] Ege Beyazit, Jeevithan Alagurajah, and Xindong Wu. 2019. Online learning from data streams with varying feature spaces. In *AAAI*, Vol. 33, 3232–3239.

[3] Albert Bifet and Ricard Gavalda. 2007. Learning from time-changing data with adaptive windowing. In *SDM*. SIAM, 443–448.

[4] Albert Bifet, Ricard Gavaldà, Geoff Holmes, and Bernhard Pfahringer. 2018. *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press. Retrieved from https://moa.cms.waikato.ac.nz/book/

[5] Olivier Cappé and Eric Moulines. 2009. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71, 3 (2009), 593–613.

[6] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

[7] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[8] Salah Ud Din, Junming Shao, Jay Kumar, Waqar Ali, Jiaming Liu, and Yu Ye. 2020. Online reliable semi-supervised learning on evolving data streams. *Information Sciences* 525 (2020), 153–171.

[9] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml

[10] Fabrizio Durante and Carlo Sempi. 2010. Copula theory: An introduction. In *Copula Theory and Its Applications*. Piotr Jaworski, Fabrizio Durante, Wolfgang Karl Härdle and Tomasz Rychlik (Eds.), Springer, 3–31.

[11] Karl B. Dyer, Robert Capo, and Robi Polikar. 2013. Compose: A semisupervised learning framework for initially labeled nonstationary streaming data. *IEEE Transactions on Neural Networks and Learning Systems* 25, 1 (2020), 12–26.

[12] Mehrdad Farajtabar, Amirreza Shaban, Hamid Reza Rabiee, and Mohammad Hossein Rohban. 2011. Manifold coarse graining for online semi-supervised learning. In *ECML-PKDD*. Springer, 391–406.

[13] Honorius Galmeanu and Razvan Andonie. 2022. Weighted incremental–decremental support vector machines for concept drift with shifting window. *Neural Networks* 152 (2022), 528–541.

[14] Andrew B. Goldberg, Ming Li, and Xiaojin Zhu. 2008. Online manifold regularization: A new learning setting and empirical study. In *ECML-PKDD*. Springer, 393–407.

[15] Bin Gu, Xiao-Tong Yuan, Songcan Chen, and Heng Huang. 2018. New incremental learning algorithm for semi-supervised support vector machine. In *SIGKDD*, 1475–1484.

[16] Yi He, Jiaxian Dong, Bo-Jian Hou, Yu Wang, and Fei Wang .2021. Online learning in variable feature spaces with mixed data. In *ICDM*. IEEE, 181–190.

[17] Yi He, Baijun Wu, Di Wu, Ege Beyazit, Sheng Chen, and Xindong Wu. 2019. Online learning from capricious data streams: a generative approach. In *IJCAI*.

[18] Yi He, Xu Yuan, Sheng Chen, and Xindong Wu. 2021. Online learning in variable feature spaces under incomplete supervision. In *AAAI*, Vol. 35, 4106–4114.

[19] Peter D. Hoff. 2007. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics* 1, 1 (2022), 265–283.

[20] Mohammad Javad Hosseini, Ameneh Gholipour, and Hamid Beigy. 2016. An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams. *Knowledge and Information Systems* 46, 3 (2016), 567–597.

[21] Bo-Jian Hou, Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. 2021. Storage fit learning with feature evolvable streams. In *AAAI*.

[22] Bo-Jian Hou, Lijun Zhang, and Zhi-Hua Zhou. 2017. Learning with feature evolvable streams. In *NeurIPS* 30 (2017).

[23] Bo-Jian Hou, Lijun Zhang, and Zhi-Hua Zhou. 2021. Prediction with unpredictable feature evolution. *IEEE Transactions on Neural Networks and Learning Systems* 33, 10 (2021), 5706–5715.

[24] Chenping Hou and Zhi-Hua Zhou. 2017. One-pass learning with incremental and decremental features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 11 (2017), 2776–2792.

[25] Chen Huang, Peiyan Li, Chongming Gao, Qinli Yang, and Junming Shao. 2019. Online budgeted least squares with unlabeled data. In *ICDM*. IEEE, 309–318.

[26] Ping Huang, Thomas Spanninger, and Francesco Corman. 2022. Enhancing the understanding of train delays with delay evolution pattern discovery: A clustering and Bayesian network approach. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 15367–15381.

[27] Botao Jiao, Yinan Guo, Dunwei Gong, and Qiuju Chen. 2022. Dynamic ensemble selection for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems* 35, 1 (2022), 1278–1291.

[28] Atsutoshi Kumagai and Tomoharu Iwata. 2018. Learning dynamics of decision boundaries without additional labeled data. In *KDD*, 1627–1636.

[29] Rui Li, Wenyin Gong, Ling Wang, Chao Lu, and Chenxin Dong. 2023. Co-evolution with deep reinforcement learning for energy-aware distributed heterogeneous flexible job shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2023).

[30] Heng Lian, John Scovil Atwood, Bojian Hou, Jian Wu, and Yi He. 2022. Online deep learning from doubly-streaming data. In *ACM Multimedia*.

[31] Han Liu, John Lafferty, and Larry Wasserman. 2009. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research* 10, 10 (2009), 2295–2328.

[32] Yanfang Liu, Xiaocong Fan, Wenbin Li, and Yang Gao. 2022. Online passive-aggressive active learning for trapezoidal data streams. *IEEE Transactions on Neural Networks and Learning Systems* 34, 10 (2022), 6725–6739.

[33] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.

[34] Guido Masarotto and Cristiano Varin. 2012. Gaussian copula marginal regression. *Electronic Journal of Statistics* 6 (2012), 1517–1549.

[35] Geoffrey J. McLachlan and Thriyambakam Krishnan. 2007. *The EM Algorithm and Extensions*. John Wiley & Sons.

[36] Yue Meng, Chunxiao Jiang, Tony Q. S. Quek, Zhu Han, and Yong Ren. 2017. Social learning based inference for crowdsensing in mobile social networks. *IEEE Transactions on Mobile Computing* 17, 8 (2017), 1966–1979.

[37] Jing Na, Yongfeng Lv, Kaiqiang Zhang, and Jun Zhao. 2020. Adaptive identifier-critic-based optimal tracking control for nonlinear systems with experimental validation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 1 (2020), 459–472.

[38] Zhengxiang Pan, Han Yu, Chunyan Miao, and Cyril Leung. 2017. Crowdsensing air quality with camera-enabled mobile devices. In *AAAI*, Vol. 31, 4728–4733.

[39] Yu-Yang Qian, Zhen-Yu Zhang, Peng Zhao, and Zhi-Hua Zhou. 2024. Learning with asynchronous labels. *ACM Transactions on Knowledge Discovery from Data* 18, 8 (2024), 186:1–186:27.

[40] Alex Rodriguez and Alessandro Laio. 2014. Clustering by fast search and find of density peaks. *Science* 344, 6191 (2014), 1492–1496.

[41] Mark Schmidt, Nicolas Le Roux, and Francis Bach. 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162, 1–2 (2017), 83–112.

[42] Christian Schreckenberger, Tim Glockner, Heiner Stuckenschmidt, and Christian Bartelt. 2020. Restructuring of Hoeffding trees for trapezoidal data streams. In *ICDM*. IEEE, 416–423.

[43] Shai Shalev-Shwartz. 2011. Online learning and online convex optimization. *Foundations and Trends in Machine Learning* 4, 2 (2011), 107–194.

[44] Junming Shao, Kai Wang, Jianyun Lu, Zhili Qin, Qiming Wangyang, and Qinli Yang. 2022. Learning evolving concepts with online class posterior probability. In *DASFAA*. Springer, 639–647.

[45] Yoram Singer and John C. Duchi. 2009. Efficient learning using forward-backward splitting. In *Advances in Neural Information Processing Systems*, Vol. 22.

[46] Aad W. Van der Vaart and Jon A. Wellner. 2000. Weak convergence. In *Weak convergence and empirical processes*, Vol. 30, no. 4, Springer, 355–373.

[47] Tal Wagner, Sudipto Guha, Shiva Kasiviswanathan, and Nina Mishra. 2018. Semi-supervised learning on data streams via temporal label propagation. In *ICML*. PMLR, 5095–5104.

[48] Di Wu, Yi He, Xin Luo, and MengChu Zhou. 2021. A latent factor analysis-based approach to online sparse streaming feature selection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 11 (2021), 6744–6758.

[49] Di Wu, Mingsheng Shang, Xin Luo, Ji Xu, Huyong Yan, Weihui Deng, and Guoyin Wang. 2018. Self-training semi-supervised classification based on density peaks of data. *Neurocomputing* 275 (2018), 180–191.

[50] Di Wu, Shengda Zhuo, Yu Wang, Zhong Chen, and Yi He. 2023. Online semi-supervised learning with mix-typed streaming features. In *AAAI*, Vol. 37, 4720–4728.

[51] SiYa Yao, Qi Kang, MengChu Zhou, Muhyaddin J. Rawa, and Aiiad Albeshri. 2022. Discriminative manifold distribution alignment for domain adaptation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 2 (2022), 1183–1197.

[52] Wangyang Ying, Dongjie Wang, Haifeng Chen, and Yanjie Fu. 2024. Feature selection as deep sequential generative learning. *ACM Transactions on Knowledge Discovery from Data* 18, 9 (2024), 221:1–221:21.

[53] Dianlong You, Jiawei Xiao, Yang Wang, Huigui Yan, Di Wu, Zhen Chen, Limin Shen, and Xindong Wu. 2023. Online learning from incomplete and imbalanced data streams. *IEEE Transactions on Knowledge and Data Engineering* 35, 10 (2023), 10650–10665.

[54] Dianlong You, Huigui Yan, Jiawei Xiao, Zhen Chen, Di Wu, Limin Shen, and Xindong Wu. 2024. Online learning for data streams with incomplete features and labels. *IEEE Transactions on Knowledge and Data Engineering* 36, 9 (2024), 4820–4834.

[55] Zhiwen Yu, Peinan Luo, Jane You, Hau-San Wong, Hareton Leung, Si Wu, Jun Zhang, and Guoqiang Han. 2015. Incremental semi-supervised clustering ensemble for high dimensional data clustering. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (2015), 701–714.

[56] Bernhard Zeisl, Christian Leistner, Amir Saffari, and Horst Bischof. 2010. On-line semi-supervised multiple-instance boosting. In *CVPR*. IEEE, 1879–1879.
[57] Qin Zhang, Peng Zhang, Guodong Long, Wei Ding, Chengqi Zhang, and Xindong Wu. 2016. Online learning from trapezoidal data streams. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2709–2723.
[58] Zhenxing Zhang and Jiuxiang Dong. 2022. A new optimization control policy for fuzzy vehicle suspension systems under membership functions online learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 5 (2022), 3255–3266.
[59] Zhen-Yu Zhang, Peng Zhao, Yuan Jiang, and Zhi-Hua Zhou. 2020. Learning with feature and distribution evolvable streams. In *ICML*. PMLR, 11317–11327.
[60] Chen Zhao, Feng Mi, Xintao Wu, Kai Jiang, Latifur Khan, and Feng Chen. 2024. Dynamic environment responsive online meta-learning with fairness awareness. *ACM Transactions on Knowledge Discovery from Data* 18, 6 (2024), 1–23.
[61] Jianwei Zhao, Zhihui Wang, and Dong Sun Park. 2012. Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing* 87 (2012), 79–89.
[62] Yuxuan Zhao and Madeleine Udell. 2020. Missing value imputation for mixed data via gaussian copula. In *SIGKDD*, 636–646.
[63] Peng Zhou, Yufeng Guo, Haoran Yu, Yuanting Yan, Yanping Zhang, and Xindong Wu. 2024. Concept evolution detecting over feature streams. *ACM Transactions on Knowledge Discovery from Data* 18, 8 (2024), 1–32.
[64] Peng Zhou, Shuai Zhang, Lin Mu, and Yuanting Yan. 2024. Online learning from capricious data streams via shared and new feature spaces. *Applied Intelligence* 54, 19 (2024), 9429–9445.
[65] Sheng-Da Zhuo, Jin-Jie Qiu, Chang-Dong Wang, and Shu-Qiang Huang. 2024. Online feature selection with varying feature spaces. *IEEE Transactions on Knowledge and Data Engineering* 36, 9 (2024), 4806–4819.

Received 5 December 2024; revised 19 March 2025; accepted 30 May 2025